PROJECT-1

Harry is a bright student. To prepare thoroughly for exams, he completes all the exercises in his book! Now that the exams are approaching fast, he is doing book exercises day and night. He writes down and keeps updating the remaining number of exercises on the back cover of each book.

Harry has a lot of books messed on the floor. Therefore, he wants to pile up the books that still have some remaining exercises into a single pile. He will grab the books one-by-one and add the books that still have remaining exercises to the top of the pile.

Whenever he wants to do a book exercise, he will pick the book with the minimum number of remaining exercises from the pile. In order to pick the book, he has to remove all the books above it. Therefore, if there are more than one books with the minimum number of remaining exercises, he will take the one which requires the least number of books to remove. The removed books are returned to the messy floor. After he picks the book, he will do all the remaining exercises and trash the book.

Since number of books is rather large, he needs your help to tell him the number of books he must remove, for picking the book with the minimum number of exercises.

Note that more than one book can have the same name.

Input

The first line contains a single integer N denoting the number of actions. Then N lines follow. Each line starts with an integer. If the integer is -1, that means Harry wants to do a book exercise. Otherwise, the integer is number of the remaining exercises in the book he grabs next. This is followed by a string denoting the name of the book.

Output

For each -1 in the input, output a single line containing the number of books Harry must remove, followed by the name of the book that Harry must pick.

Constraints

$1 < N \leq 1{,}000{,}000$
$0 \leq$ (the number of remaining exercises of each book) $< 100{,}000$
The name of each book consists of between 1 and 15 characters 'a' - 'z'.
Whenever he wants to do a book exercise, there is at least one book in the pile.

Example

```
Input:
6
```

```
9 english
6 mathematics
8 geography
-1
3 graphics
-1

Output:
1 mathematics
0 graphics
```

PROJECT-2

If there is a space to park the vehicles then some n number of vehicles are parked there and then count how many are of cars, and how many of are scooters and note the arrival time and the departure time, display the order of parked vehicles, and the color of vehicle, if he wants to take the vehicle from the parking he has to tell the number of vehicle if the number is wrong he can't able to take the vehicle from parking in that case he has to tell the car model and the color of vehicle and also the rc book of vehicle and also he has to calculate the parking fees based on time he kept.

PROJECT-3

It is winter super sale and all the shops have various offers. Suraj selected N items to buy and he is standing in the billing queue. It was then he noticed the offer "Buy two, get two". That means for every two items you buy, they give you two items for free. However, items can be of varying price, they always charge for 2 most costly items and give other 2 as free. For example, if the items cost 1, 1, 2, 2, then you have to pay 4 and take all 4 items.

Suraj is busy reordering his items to reduce the total price he has to pay. He can separate the items and get them on different bills if needed. Can you tell me what is the least price Suraj has to pay to buy all the N items?

Input

The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows. First line of each test case has single integer N. Second line of each test case has N space separated integers, which are the costs of items Suraj want to buy.

Output

For each test case, output a single line containing the required answer.

Constraints

- **$1 \leq T \leq 1000$**
- **$1 \leq N \leq 1000$**

- **1 ≤ Cost of items ≤ 1000**

Example

```
Input:
3
4
1 1 2 2
2
10 200
7
1 1 10 2 2 2 1

Output:
4
210
14
```

Explanation

Example case 1

Suraj pays for 2 costly items and gets other 2 for free.

Example case 2

Suraj has to pay for both the items, he won’t get anything for free.

Example case 3

Suraj separates the items into 2 bills. In one bill he pays 12. And in another bill he pays 2.

PROJECT-4

Pooja would like to withdraw X **$US from an ATM. The cash machine will only accept the transaction if** X **is a multiple of 5, and Pooja's account balance has enough cash to perform the withdrawal transaction (including bank charges). For each successful withdrawal the bank charges 0.50 $US. Calculate Pooja's account balance after an attempted transaction.**

Input

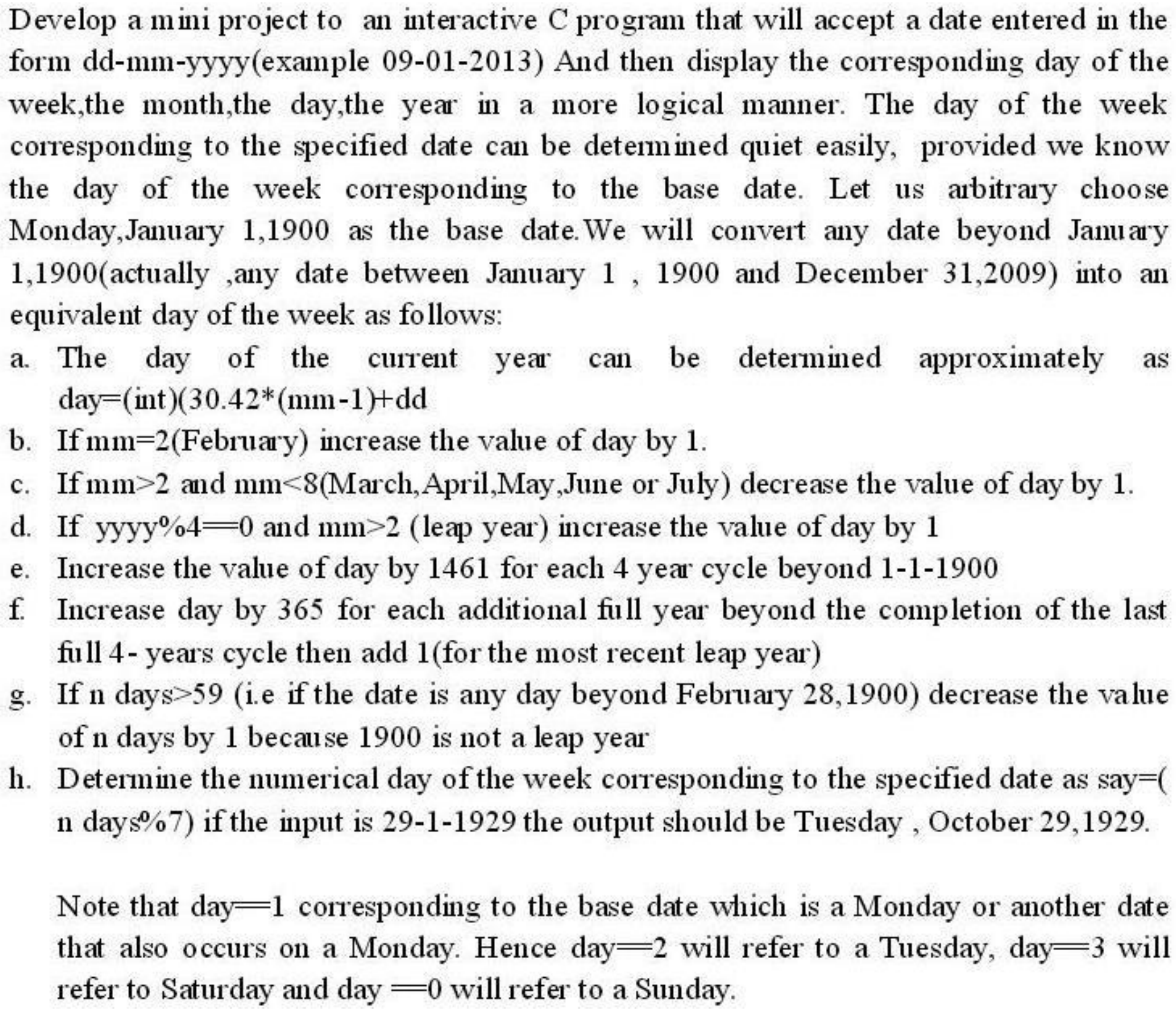
Positive integer 0 <X**<= 2000 - the amount of cash which Pooja wishes to withdraw.**

Nonnegative number 0<= Y**<= 2000 with two digits of precision - Pooja's initial account balance.**

Output

Output the account balance after the attempted transaction, given as a number with two digits of precision. If there is not enough money in the account to complete the transaction, output the current bank balance.

PROJECT-5



Develop a mini project to an interactive C program that will accept a date entered in the form dd-mm-yyyy(example 09-01-2013) And then display the corresponding day of the week,the month,the day,the year in a more logical manner. The day of the week corresponding to the specified date can be determined quiet easily, provided we know the day of the week corresponding to the base date. Let us arbitrary choose Monday,January 1,1900 as the base date.We will convert any date beyond January 1,1900(actually ,any date between January 1 , 1900 and December 31,2009) into an equivalent day of the week as follows:

a. The day of the current year can be determined approximately as day=(int)(30.42*(mm-1)+dd
b. If mm=2(February) increase the value of day by 1.
c. If mm>2 and mm<8(March,April,May,June or July) decrease the value of day by 1.
d. If yyyy%4==0 and mm>2 (leap year) increase the value of day by 1
e. Increase the value of day by 1461 for each 4 year cycle beyond 1-1-1900
f. Increase day by 365 for each additional full year beyond the completion of the last full 4- years cycle then add 1(for the most recent leap year)
g. If n days>59 (i.e if the date is any day beyond February 28,1900) decrease the value of n days by 1 because 1900 is not a leap year
h. Determine the numerical day of the week corresponding to the specified date as say=(n days%7) if the input is 29-1-1929 the output should be Tuesday , October 29,1929.

Note that day==1 corresponding to the base date which is a Monday or another date that also occurs on a Monday. Hence day==2 will refer to a Tuesday, day==3 will refer to Saturday and day ==0 will refer to a Sunday.

PROJECT-6

Pandigital products

We shall say that an n-digit number is pandigital if it makes use of all the digits 1 to n exactly once; for example, the 5-digit number, 15234, is 1 through 5 pandigital.

The product 7254 is unusual, as the identity, 39 × 186 = 7254, containing multiplicand, multiplier, and product is 1 through 9 pandigital.

Find the sum of all products whose multiplicand/multiplier/product identity can be written as a 1 through 9 pandigital.

HINT: Some products can be obtained in more than one way so be sure to only include it once in your sum.

PROJECT-7

Stone Game

A game is played with three piles of stones and two players.
At her turn, a player removes one or more stones from the piles. However, if she takes stones from more than one pile, she must remove the same number of stones from each of the selected piles.

In other words, the player chooses some N>0 and removes:

- **N stones from any single pile; or**
- **N stones from each of any two piles (2N total); or**
- **N stones from each of the three piles (3N total).**

The player taking the last stone(s) wins the game.

A *winning configuration* is one where the first player can force a win.
For example, (0,0,13), (0,11,11) and (5,5,5) are winning configurations because the first player can immediately remove all stones.

A *losing configuration* is one where the second player can force a win, no matter what the first player does.
For example, (0,1,2) and (1,3,3) are losing configurations: any legal move leaves a winning configuration for the second player. Consider all losing configurations (x_i,y_i,z_i) where $x_i \leq y_i \leq z_i \leq 100$. We can verify that $\Sigma(x_i+y_i+z_i) = 173895$ for these.

Find $\Sigma(x_i+y_i+z_i)$ where (x_i,y_i,z_i) ranges over the losing configurations
with $x_i \leq y_i \leq z_i \leq 1000$.

PROJECT-8

Sudoku is a logic-based, combinatorial number-placement puzzle. The objective is to fill a 9×9 grid so that each column, each row, and each of the nine 3×3 boxes (also called blocks or regions) contains the digits from 1 to 9 only one time each. The puzzle setter provides a partially completed grid." The rules for an N2 X N2sudoku are as follows :

The board is consists of N2 rows and N2 columns.

Numbers between 1 and N2(inclusive) are to be filled in each row such that :

All numbers in each row are distinct.

All numbers in each column are distinct.

All numbers in the sub-matrix having rows from (i*N + 1) to(i + 1)*N, and columns from (j*N + 1) to (j + 1)*N both inclusive, should be distinct. 0 <= i,j<= N-1. Rows and columns are 1 indexed. Each such sub-matrix is called a "box" or "region".

For this problem, you are required to solve a general N2X N2sudoku puzzle. Given a partially filled Sudoku board, you have to fill it in as "perfect" a manner as possible.

PROJECT-9

Chef and Operators

Chef has just started Programming; he is in first year of Engineering. Chef is reading about Relational Operators.

Relational Operators are operators which check relationship between two values. Given two numerical values A and B you need to help chef in finding the relationship between them that is,

- **First one is greater than second or,**
- **First one is less than second or,**
- **First and second one are equal.**

Input

First line contains an integer T, which denotes the number of test cases. Each of the T lines contains two integers A and B.

Output

For each line of input produce one line of output. This line contains any one of the relational operators

'<' , '>' , '='.

Constraints

- **1 ≤ T ≤ 10000**
- **1 ≤ A, B ≤ 1000000001**

Example

```
Input:
3
10 20
20 10
10 10

Output:
<
>
=
```

PROJECT-10

Chef and Two Strings

Chef has found two very old sheets of paper, each of which originally contained a string of lowercase Latin letters. The strings on both the sheets have equal lengths. However, since the sheets are very old, some letters have become unreadable.

Chef would like to estimate the *difference* between these strings. Let's assume that the first string is named S1, and the second S2. The unreadable symbols are specified with the question mark symbol `'?'`. The *difference* between the strings equals to the number of positions i, such that $S1_i$ is not equal to $S2_i$, where $S1_i$ and $S2_i$ denote the symbol at the i the position in S1 and S2, respectively.

Chef would like to know the minimal and the maximal difference between the two strings, if he changes all unreadable symbols to lowercase Latin letters. Now that you're fully aware of Chef's programming expertise, you might have guessed that he needs you help solving this problem as well. Go on, help him!

Input

The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows.

The first line of a test case contains a string S1.

The second line of a test case contains a string S2.

Both strings consist of lowercase Latin letters and question marks in places where the symbols are unreadable.

Output

For each test case, output the minimal and the maximal difference between two given strings separated with a single space.

Constraints

- 1 ≤ T ≤ 100
- 1 ≤ |S1|, |S2| ≤ 100
- Subtask 1 (25 points): |S1| = 1
- Subtask 2 (10 points): neither S1 nor S2 contains unreadable symbols
- Subtask 3 (65 points): 1 ≤ |S1|, |S2| ≤ 100

Example

```
Input:
3
a?c
??b
???a
???a
?abac
aba?w

Output:
1 3
0 3
3 5
```

Explanation

Example case 1. You can change the question marks in the strings so that you obtain S1 = `abc` and S2 = `abb`. Then S1 and S2 will differ in one position. On the other hand, you can change the letters so that S1 = `abc` and S2 = `bab`. Then, the strings will differ in all three positions.

Example case 2. Change the question marks this way: S1 = `dcba`, S2 = `dcba`, then the strings will differ in 0 positions. You can also change the question marks so that S1 = `aaaa`, S2 = `dcba`, then the strings will differ in 3 positions.

Example case 3. Change the question marks this way: S1 = `aabac`, S2 = `abaaw`, then the strings will differ in 3 positions. Then, change the question marks this way: S1 = `xabac`, S2 = `abayw`, then they will differ in 5 positions.

PROJECT-11

The Chef likes to stay in touch with his staff. So, the Chef, the head server, and the sous-chef all carry two-way transceivers so they can stay in constant contact. Of course, these transceivers have a limited range so if two are too far apart, they cannot communicate directly.

The Chef invested in top-of-the-line transceivers which have a few advanced features. One is that even if two people cannot talk directly because they are out of range, if there is another transceiver that is close enough to both, then the two transceivers can still communicate with each other using the third transceiver as an intermediate device.

There has been a minor emergency in the Chef's restaurant and he needs to communicate with both the head server and the sous-chef right away. Help the Chef determine if it is possible for all three people to communicate with each other, even if two must communicate through the third because they are too far apart.

Input

The first line contains a single positive integer T ≤ 100 indicating the number of test cases to follow. The first line of each test case contains a positive integer R ≤ 1,000 indicating that two transceivers can communicate directly without an intermediate transceiver if they are at most R meters away from each other. The remaining three lines of the test case describe the current locations of the Chef, the head server, and the sous-chef, respectively. Each such line contains two integers X,Y (at most 10,000 in absolute value) indicating that the respective person is located at position X,Y.

Output

For each test case you are to output a single line containing a single string. If it is possible for all three to communicate then you should output "yes". Otherwise, you should output "no".

To be clear, we say that two transceivers are close enough to communicate directly if the length of the straight line connecting their X,Y coordinates is at most R.

Example

```
Input:
3
1
0 1
0 0
1 0
2
0 1
0 0
1 0
```

```
2
0 0
0 2
2 1


Output:
yes
yes
no
```

PROJECT-12

Teddy and Tracy like to play a game based on strings. The game is as follows. Initially, Tracy writes a long random string on a whiteboard. Then, each player starting with Teddy makes turn alternately. Each turn, the player must erase a contiguous substring that exists in the dictionary. The dictionary consists of *N* words.

Of course, the player that can't erase any substring in his turn loses the game, and the other player is declared the winner.

Note that after a substring R is erased, the remaining substring becomes separated, i.e. they cannot erase a word that occurs partially to the left of R and partially to the right of R.

Determine the winner of the game, assuming that both players play optimally.

Input

The first line contains a single integer *T*, the number of test cases. *T* test cases follow. The first line of each test case contains a string *S*, the string Tracy writes on the whiteboard. The next line contains a single integer *N*. *N* lines follow. The *i*-th line contains a single string w_i, the *i*-th word in the dictionary.

Output

For each test case, output a single line containing the name of the winner of the game.

Example

```
Input:
3
codechef
2
code
chef
foo
1
bar
mississippi
4
```

```
ssissi
mippi
mi
ppi

Output:
Tracy
Tracy
Teddy
```

Constraints

- **1 <= *T* <= 5**
- **1 <= *N* <= 30**
- **1 <= |*S*| <= 30**
- **1 <= |w_i| <= 30**
- ***S* and w_i contain only characters 'a'-'z'**

PROJECT-13

You are given a 2D array with dimensions 6*6. An hourglass in an array is defined as a portion shaped like this:

```
a b c
  d
e f g
```

For example, if we create an hourglass using the number 1 within an array full of zeros, it may look like this:

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

Actually, there are many hourglasses in the array above. The three leftmost hourglasses are the following:

```
1 1 1     1 1 0     1 0 0
  1         0         0
1 1 1     1 1 0     1 0 0
```

The sum of an hourglass is the sum of all the numbers within it. The sums for the hourglasses above are 7, 4, and 2, respectively.

In this problem, you have to *print the largest sum among all the hourglasses* in the array.

Input Format

There will be exactly 6 lines of input, each containing 6 integers separated by spaces. Each integer will be between -9 and 9, inclusively.

Output Format

Print the answer to this problem on a single line.

Sample Input

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 2 4 4 0
0 0 0 2 0 0
0 0 1 2 4 0
```

Sample Output

19

Explanation

The hourglass possessing the largest sum is:

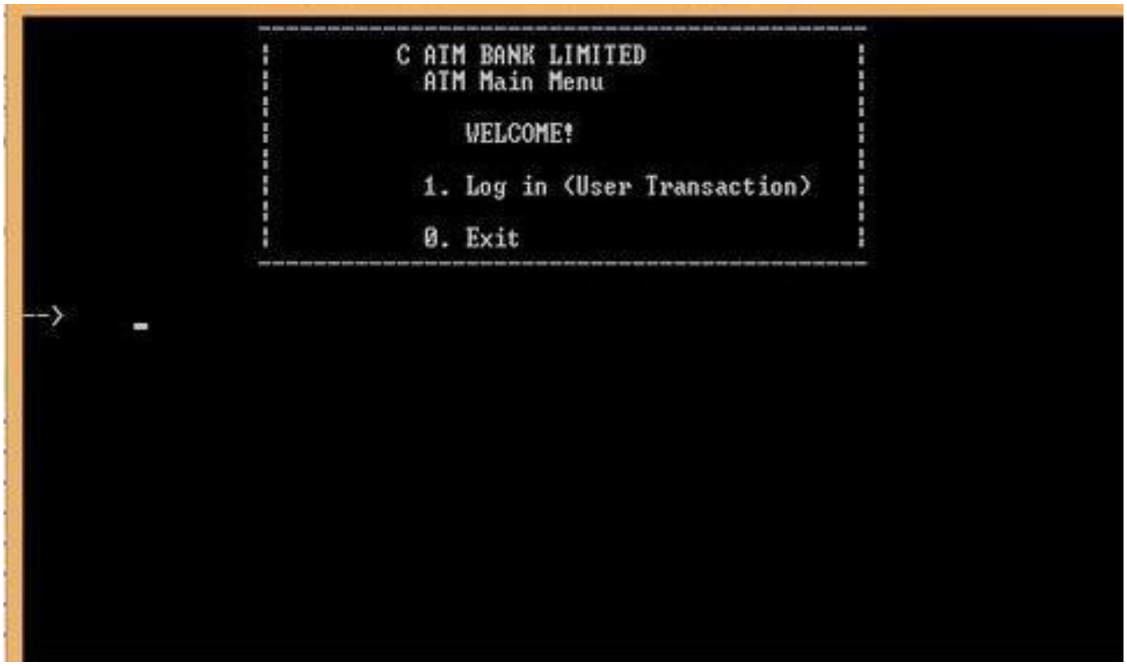
```
2 4 4
  2
1 2 4
```

PROJECT-14

ATM APPLICATION

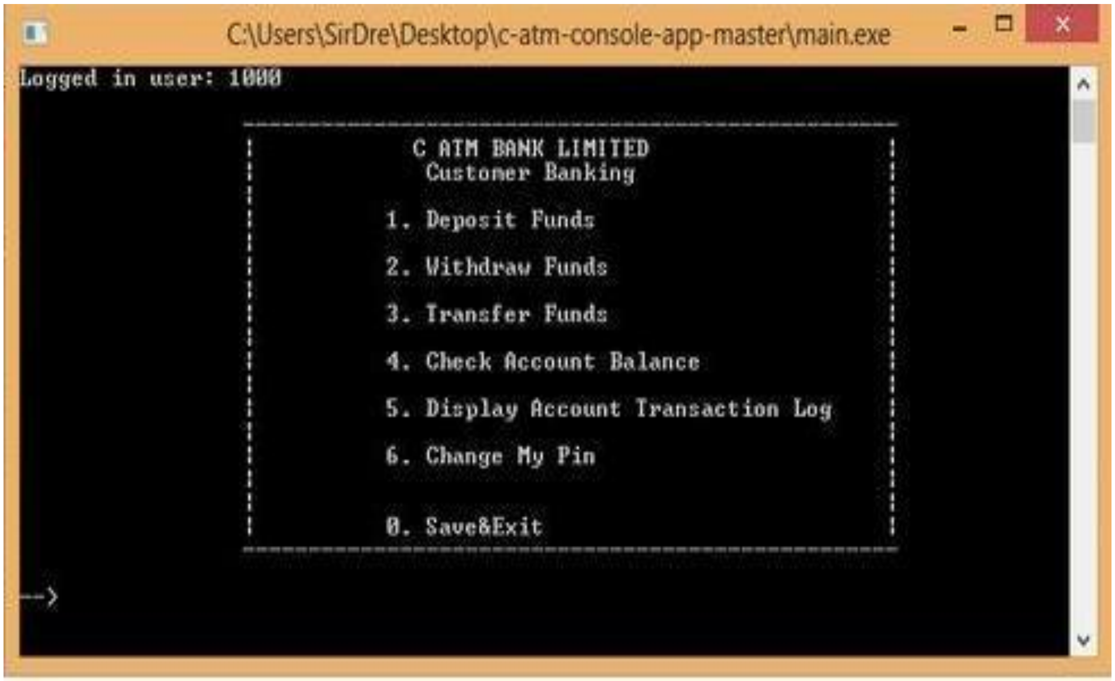
The following c console application is a simple concept of how ATM / ABM would function. Basically, the console app manipulates data using sequential and random access files by giving the users options to do so as if it's a real-time banking app.

- **Login (User Transactions)** – If selected, the user is asked for their account credentials (this is analogous to swiping the card) a/c number and pin. Once entered and successfully log-in the User Transactions menu is displayed.
- **Exit** – Ends the application.



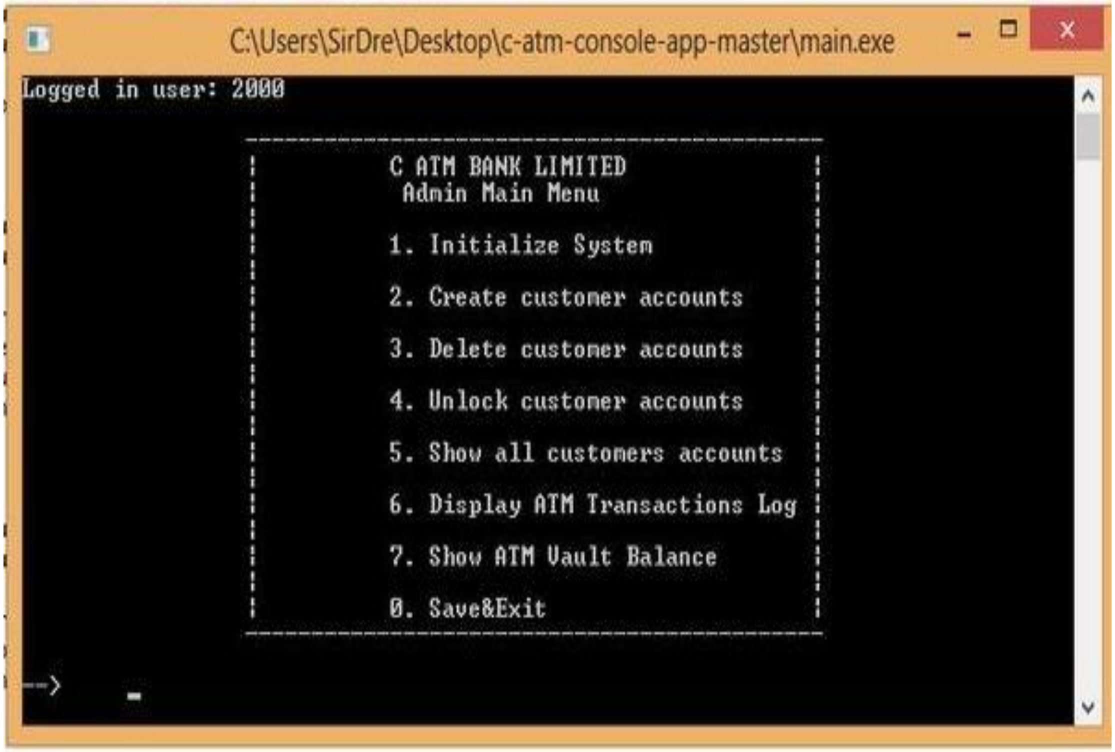
```
 ----------------------------------------------
 |           C ATM BANK LIMITED               |
 |             ATM Main Menu                  |
 |                                            |
 |               WELCOME!                     |
 |                                            |
 |            1. Log in (User Transaction)    |
 |                                            |
 |            0. Exit                         |
 ----------------------------------------------

-->      _
```

- **Customer Log-in** - If log-in with a customer a/c number then shown below are your options.



```
Logged in user: 1000

              ------------------------------------------------
              |             C ATM BANK LIMITED               |
              |              Customer Banking                |
              |           1. Deposit Funds                   |
              |           2. Withdraw Funds                  |
              |           3. Transfer Funds                  |
              |           4. Check Account Balance           |
              |           5. Display Account Transaction Log |
              |           6. Change My Pin                   |
              |                                              |
              |           0. Save&Exit                       |
              ------------------------------------------------

-->
```

- **Administrative log-in - If you are login with an admin a/c then shown below are your options.**



C:\Users\SirDre\Desktop\c-atm-console-app-master\main.exe

```
Logged in user: 2000

                 ---------------------------------------------
                 |           C ATM BANK LIMITED               |
                 |            Admin Main Menu                 |
                 |                                            |
                 |           1. Initialize System             |
                 |                                            |
                 |           2. Create customer accounts      |
                 |                                            |
                 |           3. Delete customer accounts      |
                 |                                            |
                 |           4. Unlock customer accounts      |
                 |                                            |
                 |           5. Show all customers accounts   |
                 |                                            |
                 |           6. Display ATM Transactions Log  |
                 |                                            |
                 |           7. Show ATM Vault Balance        |
                 |                                            |
                 |           0. Save&Exit                     |
                 ---------------------------------------------

-->     _
```

PROJECT-15

There are N cities in Byteland numbered 1 to N with city 1 as the capital. The road network of Byteland has a tree structure. All the roads are bi-directional. Alice and Bob are secret agents working for the Bytelandian Security Agency (BSA). Each secret agent is assigned a random city each month, where he/she must be stationed for the complete month. At the end of each month, they come to the capital to submit their reports to the head of BSA. They always take the shortest path from their city to the capital. If Alice is assigned city A and Bob is assigned city B then they meet at a city C which is common to both their routes to the capital and then travel together from C to the capital.

Alice and Bob wish to spend maximum time together in any trip. So for any pair of assigned cities (A,B) they meet at a city C such that C is the farthest city from the capital and appears in the shortest path from capital to A and capital to B. Since this happens each month they compute this for each pair of assigned cities (A,B) and store it in a matrix M, where M[A][B] = C, the city where they meet.

The importance of a city C(according to Alice and Bob), Im(C) is defined as the number of times C appears in their matrix M. Alice and Bob are interested in finding the importance of each city. Since this output can be large, output the sum S defined as S = (sum i=1 to N) i*Im(i) modulo 1000000007.

Input

First line of input contains an integer t (t<=25), the number of test cases. Each test case starts with an integer N (1<=N<=10000), the number of cities
The next N-1 lines contain two space separated integers u v (1<=u,v<=N) denoting a road from u to v.

Output

For each test case output the required sum S

Example

```
Input:
3
5
1 2
1 3
2 4
2 5
3
1 2
1 3
1

Output:
41
12
1

Explanation
For the first test case, the matrix M is:
1 1 1 1 1
1 2 1 2 2
1 1 3 1 1
1 2 1 4 2
1 2 1 2 5

and the corresponding importance array is: 15 7 1 1 1
so the required sum is 1*15 + 2*7 + 3*1 + 4*1 + 5*1 = 41

For the second test case, the matrix M is:

1 1 1

1 2 1

1 1 3

and so the Importance array is: 7 1 1

So the required sum is 1*7 + 2*1 + 3*1 = 12
```

```
For the third test case, there is only one city, so the Matrix M just has one
entry 1, so S = 1
```

PROJECT-16

How many ways are there to place a black and a white knight on an N * M chessboard such that they do not attack each other? The knights have to be placed on different squares. A knight can move two squares horizontally and one square vertically, or two squares vertically and one square horizontally. The knights attack each other if one can reach the other in one move.

Input :

The first line contains the number of test cases T. Each of the next T lines contains two integers N and M.

Output :

Output T lines, one for each test case, each containing the required answer for the corresponding test case.

```
Sample Input:
3
2 2
2 3
4 5

Sample Output:
12
26
312

Constraints:
1 <= T <= 10000
1 <= N,M <= 100000
```

PROJECT-17

Chef's younger brother is in town. He's a big football fan and has a very important match to watch tonight. But the Chef wants to watch the season finale of Master Chef which will be aired at the same time. Now they don't want to fight over it like they used to when they were little kids. They want to decide it in a fair way. So they agree to play a game to make a decision. Their favourite childhood game!

The game consists of C boards. Each board i is a grid of dimension ni x mi.

Rules of the game:

- A coin is placed at (1,1) on every board initially.

- Each one takes a turn alternatively.

- In one turn, a player can choose any one board and move a coin from a cell (i,j) to one of the following cells:

(i+1,j) OR (i+2,j) OR (i,j+1) OR (i,j+2) OR (i+1,j+1) OR (i+2,j+2).

- A coin cannot be moved out of the board at any point during the game.

- A coin cannot be moved once it reaches the cell (n,m) where n and m are the dimensions of the board of that coin.

- A player MUST make one valid move.

- The player who makes the last move gets to watch TV.

Both of them are passionate about their interests and want to watch their respective shows. So they will obviously make optimal moves in every turn. The Chef, being the elder brother, takes the first turn.

Your task is to predict which show they will be watching tonight.

Input:

The first line of input contains a single integer T, the number of test cases. T tests follow. Each test case starts with a single line containing C, the number of boards in the game. Then follow C lines: each containing 2 integers ni and mi, the dimensions of the i^{th} board.

Output:

Given the number and dimensions of boards, for each test case, output in a single line: "Master Chef" if the Chef wins or "Football" if his brother wins.

Constraints:

1<=T<=10000

1<=C<=20

2<=ni,mi<=1000

Example:

Input:

1

1

2 2

Output:

MasterChef

Explanation:

**The Chef can move the coin on the board from (1,1)->(2,2). This coin cannot be moved any further. And so, the Chef wins.
Notice that if the Chef moves it to any other valid position, i.e. either to (1,2) or (2,1) he will lose!**

PROJECT-18

Everyone knows what a square looks like. Mathematically, a square is a regular quadrilateral. This means that it has four equal sides and four equal angles (90 degree angles). One beautiful day, Johnny eagerly examined the interesting properties of squares. He did not forget you, his best friend and a talented programmer and thus made a problem about squares to challenge your programming ability. The problem is: given a set of N points in the plane, how many squares are there such that all their corners belong to this set?

Now let's show Johnny your skill!

Input

The first line contains t, the number of test cases (about 10). Then t test cases follow.

Each test case has the following form:

- **The first line contains an integer N, the number of points in the given set (4 ≤ N ≤ 500).**
- **Then N lines follow, each line contains two integers X, Y describing coordinates of a point (-50 ≤ X, Y ≤ 50).**

Output

For each test case, print in a single line the number of squares that have vertices belong to the given set.

Example

```
Input:
1
7
0 0
0 1
1 0
1 1
1 2
2 1
2 2

Output:
3

Output details:
The three squares are:
(0 0), (0 1), (1 1), (1 0)
(1 1), (1 2), (2 2), (2 1)
(0 1), (1 0), (2 1), (1 2)
```

PROJECT-19

The Little Elephant from the Zoo of Lviv likes listening to music.

There are N songs, numbered from 1 to N, in his MP3-player. The song i is described by a pair of integers B_i and L_i - the band (represented as integer) that performed that song and the length of that song in seconds. The Little Elephant is going to listen all the songs exactly once in some order.

The sweetness of the song is equal to the product of the length of that song and the number of different bands listened before (including the current playing song).

Help the Little Elephant to find the order that maximizes the total sweetness of all N songs. Print that sweetness.

Input

The first line of the input contains single integer T, denoting the number of test cases. Then T test cases follow. The first line of each test case contains single integer N, denoting the number of the songs. The next N lines describe the songs in the MP3-player. The i-th line contains two space-sparated integers B_i and L_i.

Output

For each test, output the maximum total sweetness.

Constraints

- **$1 \leq T \leq 5$**
- **$1 \leq N \leq 100000$ (10^5)**
- **$1 \leq B_i, L_i \leq 1000000000$ (10^9)**

Example

```
Input:
2
3
1 2
2 2
3 2
3
2 3
1 2
2 4

Output:
12
16
```

Explanation

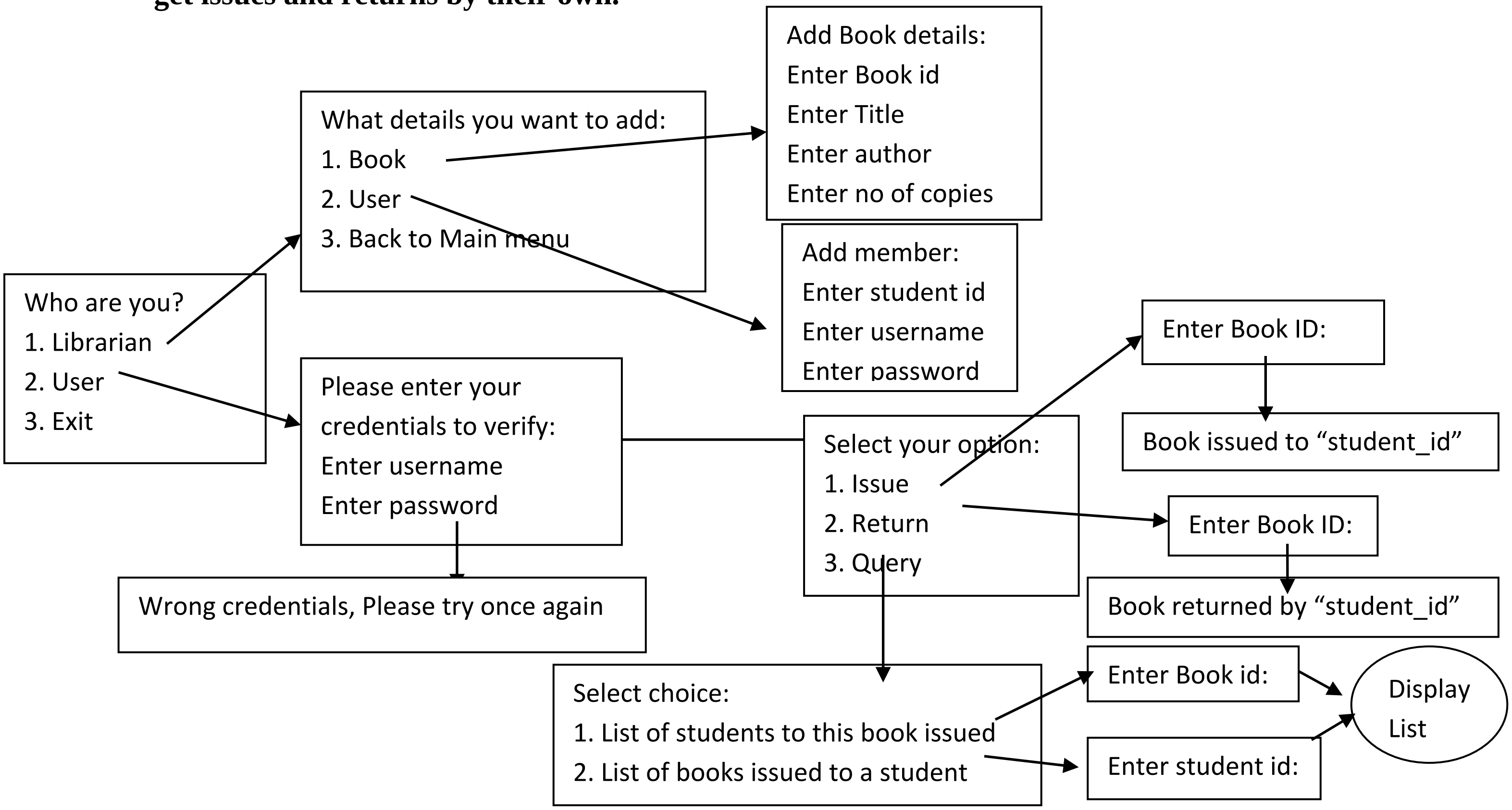
**In the first sample: if he listens the songs in given order, then
B_1=1, L_1=2: the sweetness = 2 * 1 = 2
B_2=2, L_2=2: the sweetness = 2 * 2 = 4
B_3=3, L_3=2: the sweetness = 2 * 3 = 6
So the total sweetness is 12. In this case, you can check the total sweetness does not depend on the order of the songs.**

**In the second sample: if he listens the songs in given order, then
B_1=2, L_1=3: the sweetness = 3 * 1 = 3
B_2=1, L_2=2: the sweetness = 2 * 2 = 4
B_3=2, L_3=4: the sweetness = 4 * 2 = 8
So the total sweetness is 15. However, he listens the song 2 firstly, then
B_2=1, L_2=2: the sweetness = 2 * 1 = 2
B_1=2, L_1=3: the sweetness = 3 * 2 = 6
B_3=2, L_3=4: the sweetness = 4 * 2 = 8
So the total sweetness is 16, and it is the maximum total sweetness.**

PROJECT-20

Library management system (LMS)

Our KL University wants to provide a Library management system (LMS) interface to the students and staff for the purpose of self issue and returns. If any user (student or staff) wants to take book from library, he must interact with Library management system by providing the credentials (username and password) of user. Then, LMS Allows the user to get issues and returns by their own.



Our librarian maintains the details of books that are available in library, details (credentials) of users that are already member of library. Here, we need two structures to maintain these details. Library member structure contains student_id, username, password. The Book structure contains Book_ID, Title, author, number of copies, Book issued to (is a student_id who has taken copy of that book) .

The interface asks the user to enter his username and password. It verifies the credentials of that particular user. If the entered credentials are correct, it will display two options 1. Issues 2. Returns 3. Query. Otherwise, it will display a message "wrong credentials, Please try once again". User selects first option to get issue. The book will be issued to that student by entering Book id. If selects second option, student can return the book by entering book id. The option three provides a querying facility to enquire about list of books issued to a particular student and a particular book issued a list of students.

PROJECT-21

EMPLOYEE MANAGEMENT SYSTEM

In this project, we maintain the details of all employees and their children using nesting of structures. Consider each employee has four children and all are studying same number of subjects in same class. These are the structure members of Employee, Children, and Subjects.

Employee	**Children**	**Subjects**
ID	name	sub1
Name	age	sub2
Age	gender	sub3
Gender	struct subjects	total
Salary		
Struct children		

Here, You need to

1. List all employees whose children got 75% of total marks.

2. List children names of employees whose salary more than the average salary of all employees

3. Count the number of male and female children of all employees.

4. Sort children of all employees by their names and print it.

PROJECT-22

HOTEL MANAGEMENT SYSTEM

Develop an application for hotel management system with the following modules using structures, pointers to structure variables, passing structure pointers to function.

1. Get availability

2. Features of room

3. Room allocation

4. Show customer details

5. Room deallocation

6. Restaurant

7. Billing.

Make your own assumptions for this project, design and implement hotel management system.

PROJECT-23

Project Description: By Using this Program User can able to do different following operations using Arrays. 1. Insertion, 2. Deletion, 3. Sorting, 4. Searching, 5. Update, 6. Retrieve, 7. Merging, 8. Append, 9. Exit. This Menu Driven Program will enhance the array operations capability in c programming. In this program User can insert, delete, sort, search, update, retrieve, merge, append the elements in array with respect to the user choice.

PROJECT-24

Mini project "Calendar Application" is also a simple project made using C. It uses many windows properties to make it colorful, for example, to indicate the vacation, it uses the red foreground color. The calendar can be used for two purposes. First to see the date and month as usual calendars and second to find out the day corresponding to given date. Some of the silent features of the project are

- **It uses various windows properties to make the program colorful although it has lack of graphics.**
- **It entirely uses C codes, code is written in simple manner with lots of comments Important notes can be added.**
- **The date with such notes appears different than others with red background color.**
- **The months can be navigated using arrow keys.**

PROJECT-25

MODERN PERIODIC TABLE

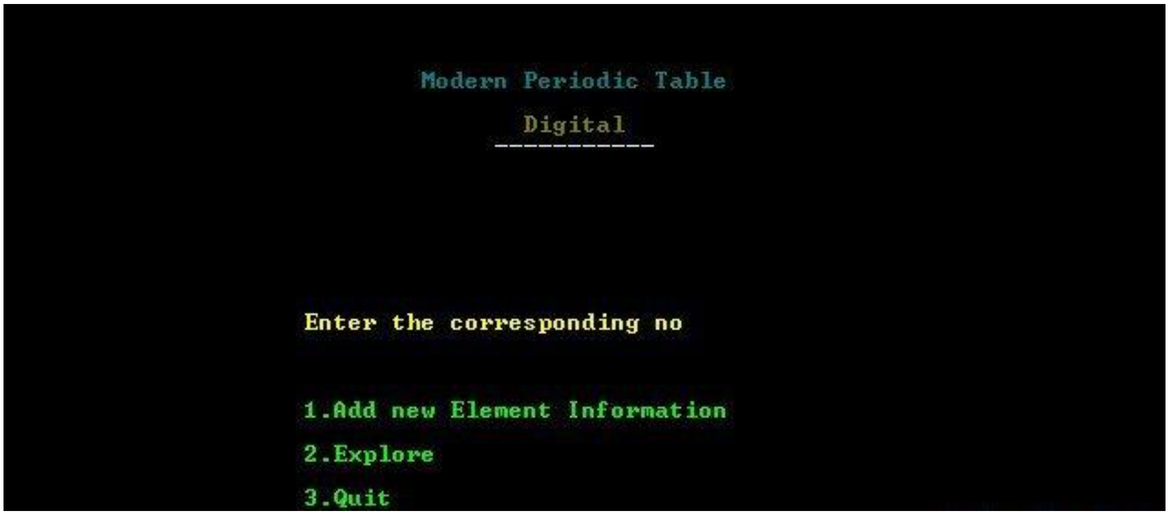
This project will help you to understand file handling in C i.e. creating a file and accessing the stored data in the file, modifying and removing the stored data. It will also help you to understand the use of functions as well as different parameters of C programming language.

The key **features** of Modern Periodic Table mini project in C are briefly described below:

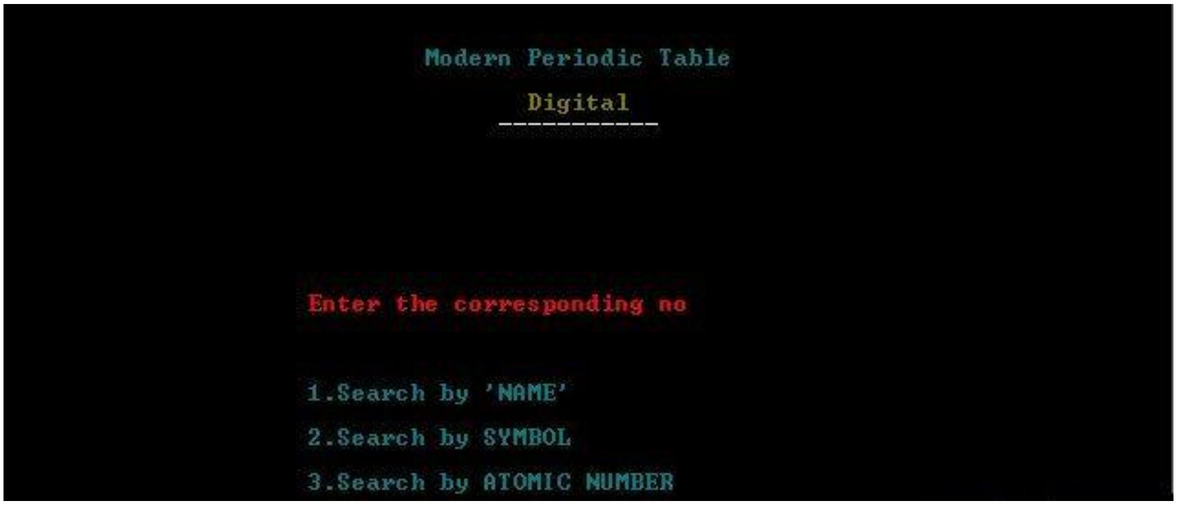
- **Storage of Element Information:** In the project, you can add any new element with its name, symbol, atomic number, atomic weight and its some important properties. When new element information is to be added to this Modern Periodic Table, you have to enter 1 in the main menu and input information in given format. This information is stored in file created on the hard disk of computer by program itself.

- **Exploration of element Information:** Another main function of this project is to explore or to display the stored information. You can search an element by using any of the following method:
    1. By name of element
    2. By symbol of element
    3. By atomic number of element
    4. By atomic weight of element

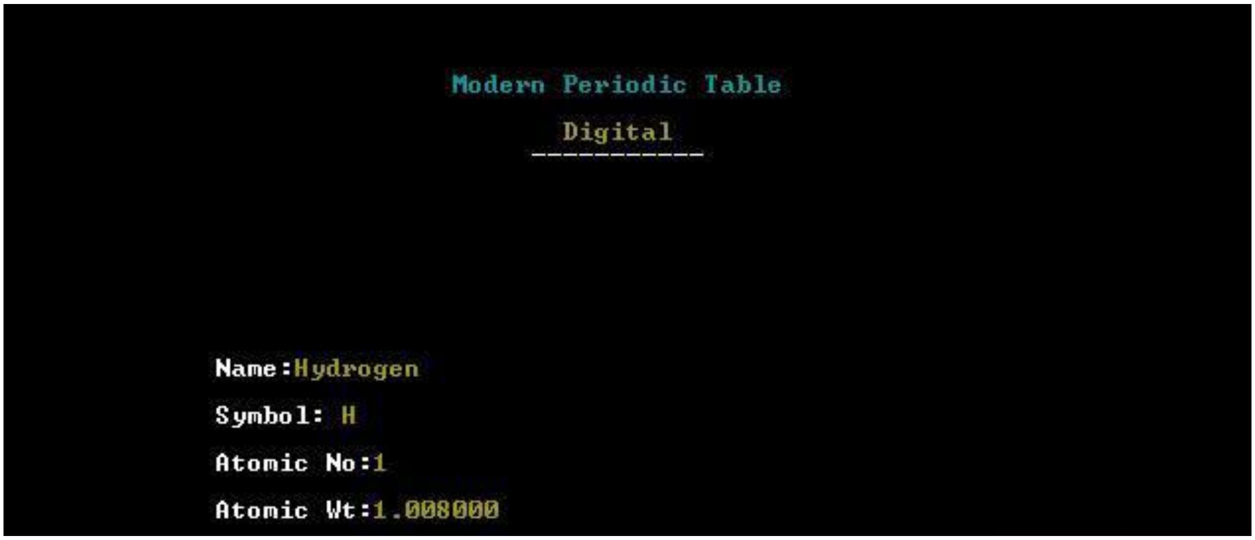
- If you press 3 in the main menu, the program will be terminated.



```
                    Modern Periodic Table

                           Digital
                         ___________




           Enter the corresponding no


           1.Add new Element Information

           2.Explore

           3.Quit
```



```
                    Modern Periodic Table

                           Digital
                         ___________




           Enter the corresponding no


           1.Search by 'NAME'

           2.Search by SYMBOL

           3.Search by ATOMIC NUMBER
```



```
                    Modern Periodic Table

                          Digital
                        -----------




     Name:Hydrogen

     Symbol: H

     Atomic No:1

     Atomic Wt:1.008000
```

PROJECT-26

CRICKET SCORE SHEET

First of all the project displays the welcome screen and the screen fades up to display the main menu. The main menu comprises three options namely:

-> New Score Sheet

-> View Score Sheet

-> Exit

If ‘1’ is entered, Cricket Score Sheet project asks for the name of new score sheet. When the file is created, a message is displayed on the screen.

Then, the score sheet appears on screen in which the user has to input the following information:

Competition

Venue

Match between and versus

Toss winner team

Elected choice of toss winner

Inning and date

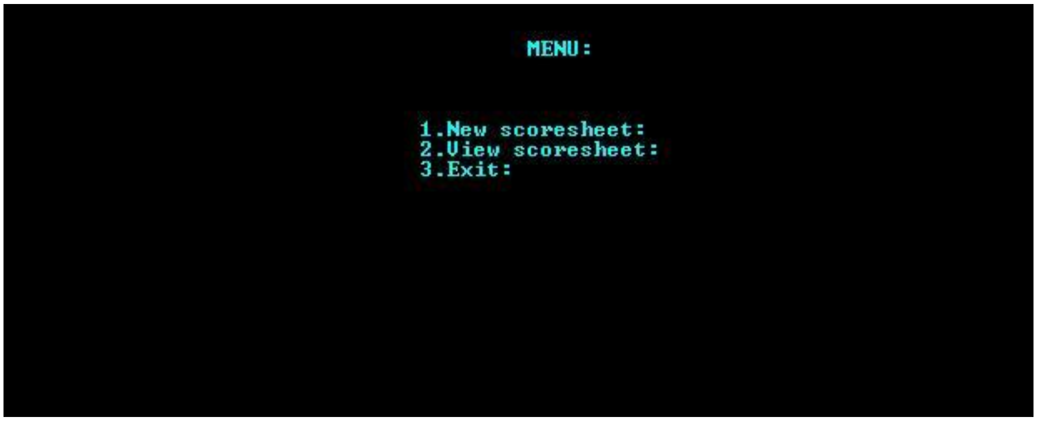
Name of batsman and run hit by each of them

Name of bowler and run given by each blower

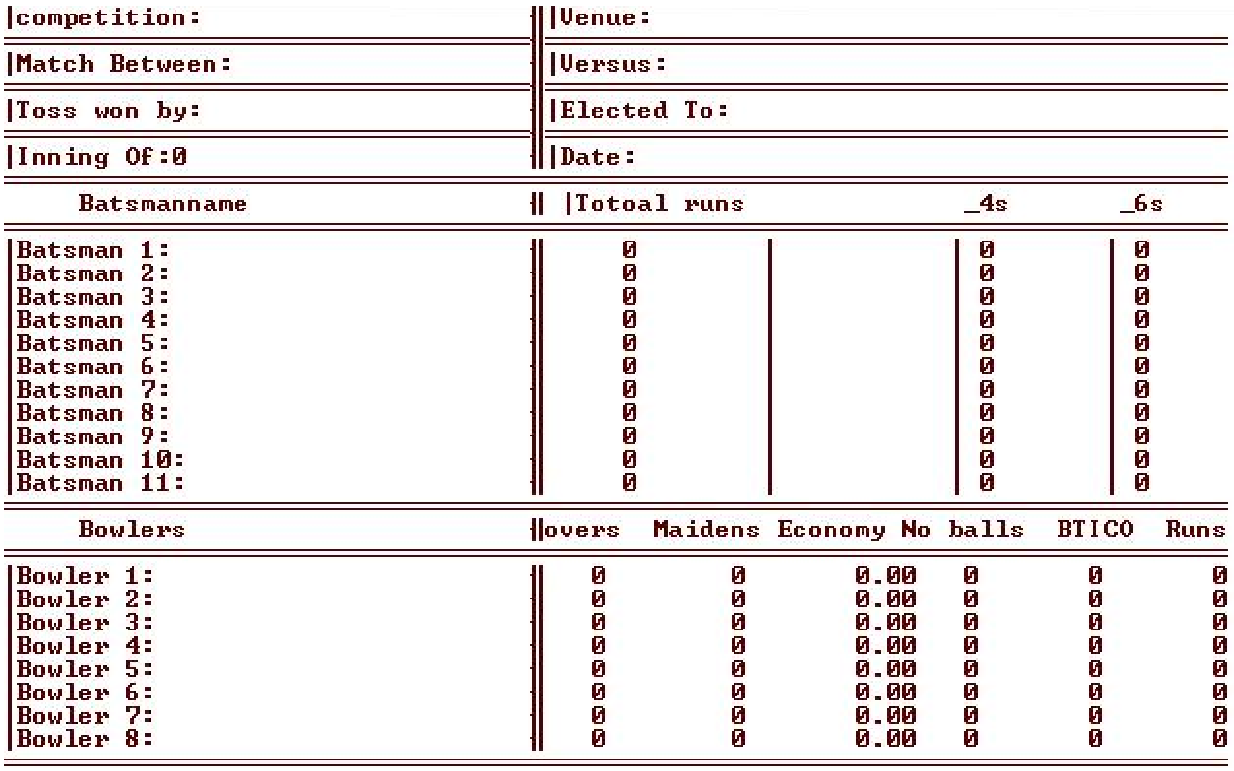
After inputting these data, the program asks user to input ‘e’ to edit the data and ‘c’ to continue.

If the user inputs ‘2’ in the main menu, the program asks for the name of file. If the file is found, it is displayed. Otherwise, error message is printed on the screen.

The third option in main menu is exit. If ‘3’ is entered in the main menu, the Cricket Score Sheet project terminates.



```
                MENU:


        1.New scoresheet:
        2.View scoresheet:
        3.Exit:
```

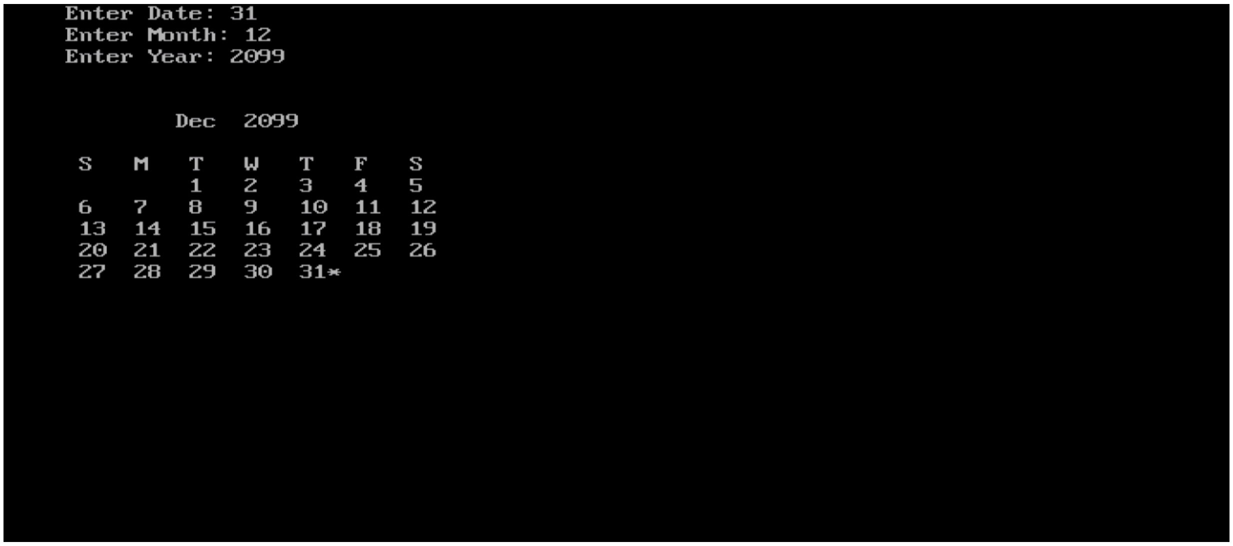


```
|competition:                   ||Venue:
==========================================================================
|Match Between:                 ||Versus:
==========================================================================
|Toss won by:                   ||Elected To:
==========================================================================
|Inning Of:0                    ||Date:
==========================================================================
    Batsmanname                 || |Totoal runs            _4s       _6s
==========================================================================
|Batsman 1:                     ||    0      |          |  0     |  0
|Batsman 2:                     ||    0      |          |  0     |  0
|Batsman 3:                     ||    0      |          |  0     |  0
|Batsman 4:                     ||    0      |          |  0     |  0
|Batsman 5:                     ||    0      |          |  0     |  0
|Batsman 6:                     ||    0      |          |  0     |  0
|Batsman 7:                     ||    0      |          |  0     |  0
|Batsman 8:                     ||    0      |          |  0     |  0
|Batsman 9:                     ||    0      |          |  0     |  0
|Batsman 10:                    ||    0      |          |  0     |  0
|Batsman 11:                    ||    0      |          |  0     |  0
==========================================================================
    Bowlers                     ||overs  Maidens Economy No balls  BTICO  Runs
==========================================================================
|Bowler 1:                      ||  0       0      0.00   0        0      0
|Bowler 2:                      ||  0       0      0.00   0        0      0
|Bowler 3:                      ||  0       0      0.00   0        0      0
|Bowler 4:                      ||  0       0      0.00   0        0      0
|Bowler 5:                      ||  0       0      0.00   0        0      0
|Bowler 6:                      ||  0       0      0.00   0        0      0
|Bowler 7:                      ||  0       0      0.00   0        0      0
|Bowler 8:                      ||  0       0      0.00   0        0      0
==========================================================================
```

PROJECT-27

The simple Project should accept the date month and year between 1900 to 2100 and should display the calendar of that particular month displaying like below:

The project should be finished without using graphics. with logic you have to accomplish the project



```
Enter Date: 31
Enter Month: 12
Enter Year: 2099


        Dec  2099

 S   M   T   W   T   F   S
         1   2   3   4   5
 6   7   8   9   10  11  12
 13  14  15  16  17  18  19
 20  21  22  23  24  25  26
 27  28  29  30  31*
```

PROJECT-28

The results from the mayor's race have been reported by each precinct as follows:

	Candidate	Candidate	Candidate	Candidate
Precinct	A	B	C	D
1	192	48	206	37
2	147	90	312	21
3	186	12	121	38
4	114	21	408	39
5	267	13	382	29

Write a program to do the following:

a. Read the raw vote totals from a data file that contains one row for each precinct.

b. Display the table with appropriate headings for the rows and columns.

c. Compute and display the total number of votes received by each candidate and the percent of the total votes cast.

d. If any one candidate received over 50% of the votes, the program should print a message declaring that candidate the winner.

e. If no candidate received 50% of the votes, the program should print a message declaring a run-off between the two candidates receiving the highest number of votes; the two candidates should be identified by their letter names.

f. For testing, run the program with the above data, and also with another data file where Candidate C receives only 108 votes in precinct 4.

PROJECT-29

One classic method for composing secret messages is called a square code. The spaces are removed from the English text and the characters are written into a square (or rectangle). For example, the sentence "If man was meant to stay on the ground god would have given us roots" is 54 characters long, so it is written into a rectangle with 7 rows and 8 columns.

```
ifmanwas
meanttos
tayonthe
groundgo
dwouldha
vegivenu
sroots
```

The coded message is obtained by reading down the columns going left to right. For example, the message above is coded as:

```
imtgdvs fearwer mayoogo anouuio ntnnlvt wttddes aohghn  sseoau
```

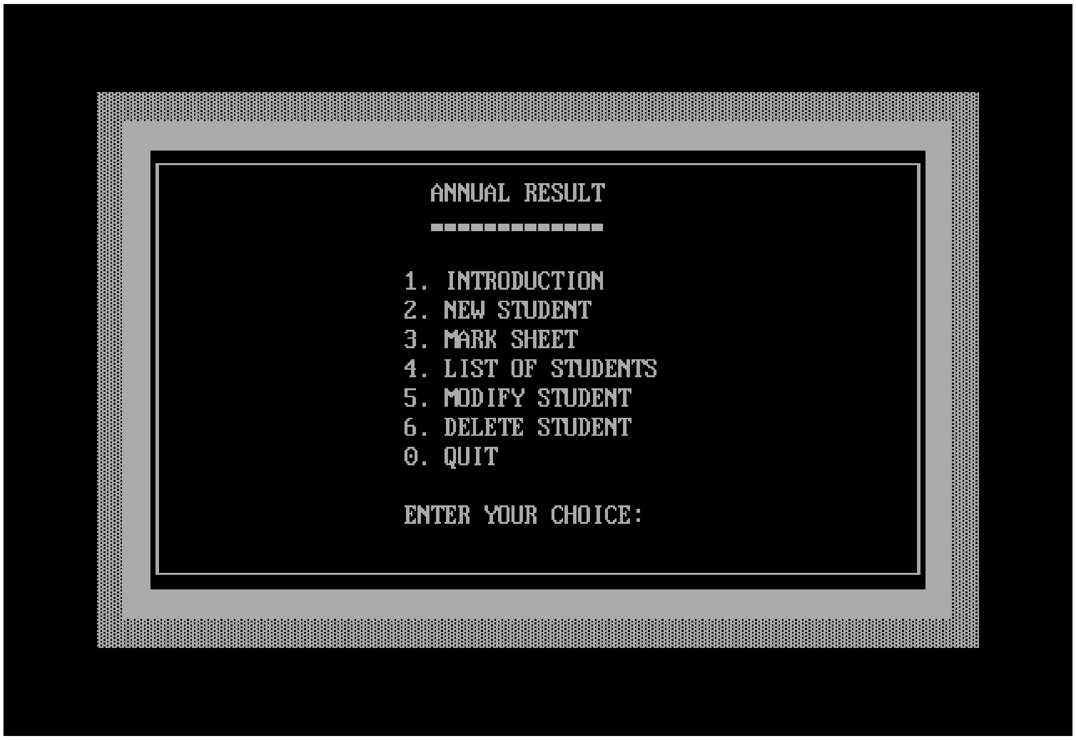
In your program, have the user enter a message in english with no spaces between the words. Have the maximum message length be 81 characters. Display the encoded message. (Watch out that no "garbage" characters are printed.) Here are some more examples:

```
   Input                                                Output
haveaniceday                                         hae and via ecy
feedthedog                                           fto ehg ee dd
chillout                                             clu hlt io
```

PROJECT – 30

Develop a Banking project in c language which will implement following features or functionality in the program.

- **Account Creation**
- **Deposit Amount**
- **Withdraw Amount**
- **View Details**
- **Foreign Exchange**
- **Exit program**



```
  ANNUAL RESULT
  ■■■■■■■■■■■■■

1.  INTRODUCTION
2. NEW STUDENT
3. MARK SHEET
4. LIST OF STUDENTS
5. MODIFY STUDENT
6. DELETE STUDENT
0. QUIT

ENTER YOUR CHOICE:
```