## GE23131-Programming Using C-2024



Status Finished Started Monday, 23 December 2024, 5:33 PM Completed Monday, 25 November 2024, 11:16 AM Duration 28 days 6 hours Question 1 Write a program that prints a simple chessboard. Correct Marked out of 3.00 Input format: F Flag question The first line contains the number of inputs T. The lines after that contain a different values for size of the chessboard Output format: Print a chessboard of dimensions size \* size. Print a Print W for white spaces and B for black spaces. Input: 2 3 5 Output: WBW BWB WBW WBWBW **BWBWB** WBWBW **BWBWB** WBWBW Answer: (penalty regime: 0 %) #include<stdio.h> 2 int main() 3. {

4

int v;

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
    int main()
 2
 3 + {
        int v;
scanf("%d",&v);
        while(v>0)
 6
 7.
             int x;
scanf("%d",&x);
 8
             if(x<0)
10
11.
             {
12
                 x=-x;
13
             char a = 'W';
14
             for(int i = 0;i<x;i++)
15
16 .
                 for(int j=0;j<x;j++)
17
18 .
                      printf("%c",a);
19
20
                      1f(a=='W')
21
                      {
                          a = 'B';
22
23
24
                      else
25
                      {
                          a ='W';
26
27
28
                 printf("\n");
29
                 if(x\%2==0)
30
31 .
32
                      if(a=='W')
                      a='B';
33
                      else
34
35
                      a='W';
36
37
38
39
40
        return 0;
41 }
```

~	2	WBW	WBW	~
•	3	BWB	BWB	
	5	MBM	WBW	
		MBMBM	WBWBW	
		BMBMB	BMBMB	
		MBMBM	WBWBW	
		BMBWB	BMBMB	
		MBMBM	WBWBW	

Question 2 Correct Marked out of 5.00

P Flag question Let's print a chessboard!

Write a program that takes input:

The first line contains T, the number of test cases

Each test case contains an integer N and also the starting character of the chessboard

**Output Format** 

Print the chessboard as per the given examples

r Flag question

The first line contains T, the number of test cases Each test case contains an integer  $\tilde{N}$  and also the starting character of the chessboard

**Output Format** 

Print the chessboard as per the given examples

Sample Input / Output

Input:

2

2 W

3 B

Output:

WB

BW

BWB

WBW

BWB

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
    int main()
 2
 3. {
        int t,d,i,i1,i2,o,z;
 5
        char c,s;
        scanf("%d",&t);
for(i=0;i<t;i++)
 6
 7
 9
            scanf("%d %c",&d,&s);
10
            for(i1=0;i1<d;i1++)
11.
                 z=(s=='W')?0:1;
12
                 o=(i1%2==z)?0:1;
13
                 for(i2=0;i2<d;i2++)
14
15.
                     c=(i2%2==o)?'W':'B';
16
17
                     printf("%c",c);
18
19
                 printf("\n");
20
            }
21
        }
22
        return 0;
23 }
```

	Input	Expected	Got	
~	2	WB	WB	~
	2 W	BW	BW	
	3 B	BMB	BMB	
		MBM	WBW	
		BMB	BMB	

Passed all tests! V

Question 3

Marked out of 7.00

Correct

F Flag question Decode the logic and print the Pattern that corresponds to given input. If N= 3 then pattern will be: 10203010011012 \*\*4050809 \*\*\*\*607 If N= 4, then pattern will be: 1020304017018019020 \*\*50607014015016 \*\*\*\*809012013 \*\*\*\*\*10011 Constraints 2 <= N <= 100 Input Format First line contains T, the number of test cases Each test case contains a single integer N Output First line print Case #i where I is the test case number In the subsequent line, print the pattern Test Case 1 3 3 5 Output Case #1 10203010011012 \*\*4050809 \*\*\*\*607 Case #2 1020304017018019020

> \*\*50607014015016 \*\*\*\*809012013

```
***6070809022023024025
****10011012019020021
*****13014017018
******15016
```

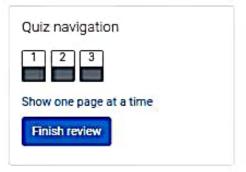
Answer: (penalty regime: 0 %)

```
#include<stdio.h>
    int main()
 2
 3 . {
        int n ,v,p,c,i,in,i1,i2,t,ti;
scanf("%d",&t);
for(ti=0;ti<t;ti++)</pre>
 4
 5
 6
 7.
             v=0;
 8
             scanf("%d",&n);
 g
10
             printf("Case #%d\n",ti+1);
             for(i=0;i<n;i++)
11
12 .
             {
                  c=0;
13
                 if(i>0)
14
15 .
16
                      for(i1=0;i1<i;i1++) print
                  }
17
18
19
             for(i1=i;i1<n;i1++)
20
21 .
22
                  if(i>0) c++;
23
                  printf("%d0",++v);
24
25
             if(i==0)
26 •
                  p = v+(v*(v-1))+1;
27
28
                 in=p;
29
30
             in=in-c;
31
             p=in;
             for(i2=i;i2<n;i2++)
32
33.
34
                  printf("%d",p++);
35
                  if(i2!=n-1) printf("0");
36
37
             printf("\n");
38
39
         }
40 }
```

	Input	Expected	Got
~	3	Case #1	Case #1
	3	10203010011012	102030100110
	4	**4050809	**4050809
	5	****607	****607
		Case #2	Case #2
		1020304017018019020	102030401701
		**50607014015016	**5060701401
		****809012013	****80901201
		******10011	******10011
		Case #3	Case #3
		102030405026027028029030	102030405026
		**6070809022023024025	**6070809022
		****10011012019020021	****10011012
		*****13014017018	******130140
		*******15016	*******1501

Passed all tests! <

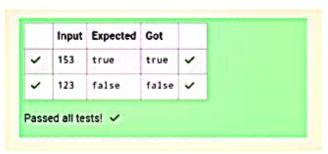
## GE23131-Programming Using C-2024



Status Finished Started Monday, 23 December 2024, 5:33 PM Completed Monday, 25 November 2024, 1:28 PM Duration 28 days 4 hours Question 1 The k-digit number N is an Armstrong number if and only if Correct the k-th power of each digit sums to N. Marked out of F Flag Given a positive integer N, return true if and only if it is an question Armstrong number. Example 1: Input: 153 Output: true Explanation: 153 is a 3-digit number, and 153 = 1\*3 + 5\*3 + 3\*3. Example 2: Input: 123 Output: false Explanation: 123 is a 3-digit number, and 123 != 1^3 + 2^3 + 3^3 = 36. Example 3:

Input:

```
Output:
true
Note:
1 <= N <= 10*8
Answer: (penalty regime: 0 %)
       #include<math.h>
   2
       #include<stdio.h>
   3
      int main()
   4 .
   5
           int n;
           scanf("%d",&n);
   6
   7
           int x=0, n2=n;
   8
           while(n2!=0)
   9
  10
               n2=n2/10;
  11
  12
  13
           int sum=0;
  14
           int n3=n,n4;
           while(n3!=0)
  15
  16
               n4=n3%10;
  17
  18
               sum = sum + pow(n4,x);
  19
               n3=n3/10;
  20
  21
           if(n==sum)
  22
           {
  23
               printf("true");
  24
           }
  25
           else
  26
           {
  27
               printf("false");
  28
           }
  29
           return 0;
  30
  31 }
```



Question 2
Correct
Marked out of 5.00
F Flag question

Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints 1<=num<=99999999 Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
 2
   int main()
3.
        int rn,n,nt=0,1=0;
5
        scanf("%d",&n);
 6
 7
            nt=n;rn=0;
 8
            while(n!=0)
 9
10
                rn=rn*10+n%10;
                n=n/10;
11
12
```

inserance.

Autower. (penalty regime. U 10)

```
#include<stdio.h>
2
   int main()
3.
 4
        int rn,n,nt=0,i=0;
5
        scanf("%d",&n);
6.
        do{
7
            nt=n;rn=0;
 8
            while(n!=0)
9.
                rn=rn*10+n%10;
10
                n=n/10;
11
12
13
            n=nt+rn;
14
15
        }while(rn!=nt||i==1);
        printf("%d",rn);
16
17
        return 0;
18 }
```

	Input	Expected	Got	
~	32	55	55	~
~	789	66066	66066	~

Question 3
Correct
Marked out of 7.00
F Flag

question

A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a program to print the nth lucky number. Example, 1st lucky number is 3, and 2nd lucky number is 4 and 3rd lucky number is 33 and 4th lucky number is 34 and so on. Note that 13, 40 etc., are not lucky as they have other numbers in it.

The program should accept a number 'n' as input and display the nth lucky number as output.

Sample Input 1:

3

Sample Output 1:

33

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

34

Sample Output 2:

33344

3

Sample Output 1:

33

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

34

Sample Output 2:

33344

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
 2
3. {
         int n=1,i=0,nt,co=0,e;
scanf("%d",&e);
while(i<e)</pre>
 4
 6
 7
 8
              nt=n;
              while(nt!=0)
 9
10.
                   co=0;
11
12
                   if(nt%10!=3&&nt%10!=4)
13
14
                        co=1;
15
                       break;
16
17
                   nt=nt/10;
18
19
              if(co==0)
20 .
              {
21
                   1++;
22
              }
23
              n++;
24
25
         printf("%d", --n);
26
27
         return 0;
28 }
```

	Input	Expected	Got	
~	34	33344	33344	~

Finish review