GE23131-Programming Using C-2024

Status	Finished
Started Monday, 23 December 2024, 5:33 PM	
Completed Monday, 23 December 2024, 2:12 PM	
Duration	3 hours 20 mins

Question 1

Correct

Marked out of 3.00

Flag question

The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.

Given a positive integer N, return true if and only if it is an 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:

Output:

false

Explanation:

123 is a 3-digit number, and 123 != $1^3 + 2^3 + 3^3 = 36$.

Example 3:

Input:

1634

Output:

true

Note:

1 <= N <= 10^8

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
    #include<math.h>
2
 3
 4
   int main()
 5 ▼
    {
 6
        int n;
 7
        scanf("%d",&n);
        int x=0, n2=n;
 8
        while(n2!=0)
 9
10 •
        {
11
             X++;
            n2=n2/10;
12
13
        int sum=0;
14
        int n3=n,n4;
15
16
        while(n3!=0)
17
```

```
n4=n3\%10;
18
19
             sum=sum+pow(n4,x);
             n3=n3/10;
20
21
         if(n==sum)
22
23 •
             printf("true");
24
25
         }
26
         else
27 •
28
             printf("false");
29
30
         return 0;
31
   }
```

	Input	Expected	Got	
~	153	true	true	~
~	123	false f	false	~

Passed all tests! <

Question 2

Correct

Marked out of 5.00

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>
 1
 2
    int main()
 3 ▼
    {
         int rn,n,nt=0,i=0;
 4
 5
         scanf("%d",&n);
 6
         do{
 7
             nt=n;rn=0;
             while(n!=0)
 8
 9 •
10
                 rn=rn*10 +n%10;
11
                 n=n/10;
12
13
             n=nt+rn;
14
```

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

Passed all tests! <

Question 3

Correct

Marked out of 7.00

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:

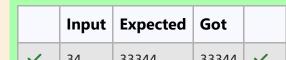
34

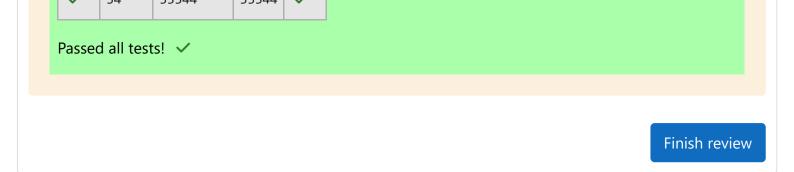
Sample Output 2:

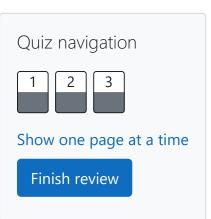
33344

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
    int main()
 2
 3 ▼
    {
 4
         int n=1,i=0,nt,co=0,e;
         scanf("%d",&e);
 5
         while(i<e)</pre>
 6
 7 🔻
         {
 8
             nt=n;
             while(nt!=0)
 9
10 •
11
                  co=0;
                  if(nt%10!=3 && nt%10!=4)
12
13 •
14
                       co=1;
15
                      break;
16
                  nt=nt/10;
17
18
19
             if(co==0)
20 🔻
21
                  i++;
22
              }
23
             n++;
24
         printf("%d",--n);
25
26
         return 0;
27
```







GE23131-Programming Using C-2024

Status	Finished
Started Tuesday, 24 December 2024, 3:42 PM	
Completed Tuesday, 24 December 2024, 4:06 PM	
Duration	24 mins 23 secs

Question **1**

Correct

Marked out of 3.00

Flag question

Write a program that prints a simple chessboard.

Input format:

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>
    int main()
 2
 3 ▼
         int n,size,i,j,count;
 4
         scanf("%d",&n);
 5
 6
         while(n--)
 7 🔻
         {
             scanf("%d",&size);
 8
 9
             count = 0;
             for(i=0;i<size;i++)</pre>
10
11 •
12
                  for(j=0;j<size;j++)</pre>
13 •
14
                      if(++count%2==1)
15
                      printf("W");
16
                      else
17
                      printf("B");
18
19
                  if(size%2==0)
20
                  count++;
21
22
                  printf("\n");
23
24
25
   }
```

	Input	Expected	Got	
~	2	WBW	WBW	~
	3	BWB	BWB	
	5	WBW	WBW	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	

Passed all tests! <

Correct Marked out of 5.00 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 Sample Input / Output Input: 2 2 W 3 B Output: WB BW **BWB WBW BWB Answer:** (penalty regime: 0 %) #include<stdio.h> 2 int main()

Question **2**

3 ▼ {

int n.

```
לוו אווד
 5
         int size,i,j,count;
         char ch,first,second;
 6
 7
         scanf("%d",&n);
 8 •
         while(n--){
             scanf("%d",&size);
 9
             scanf(" %c",&ch);
10
             if(ch == 'W')
11
12 •
13
                  first= 'W';
                  second= 'B';
14
15
             }
             else
16
17 •
                  first = 'B';
18
                  second = 'W';
19
20
21
             count = 0;
22
             for(i=0;i<size;i++)</pre>
23 🔻
             {
24
                  for(j=0;j<size;j++)</pre>
25 •
                  {
                      if(++count%2==1)
26
                      printf("%c", first);
27
28
                      else
                      printf("%c", second);
29
30
                  if(size%2 == 0)
31
32
                  count++;
33
                  printf("\n");
34
35
36
   }
```

	Input	Expected	Got	
~	2 2 W	WB BW	WB BW	~
	3 B	BWB	BWB	
		WBW BWB	WBW BWB	

Passed all tests! ✓

Question **3**

Correct

Marked out of 7.00

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

```
Output

Case #1

10203010011012

**4050809

****607

Case #2

1020304017018019020

**50607014015016

****809012013

******10011

Case #3

102030405026027028029030

**6070809022023024025

****10011012019020021

******13014017018
```

5

Answer: (penalty regime: 0 %)

******15016

```
#include<stdio.h>
 2
    int main()
 3 ▼
 4
         int n,row,col,opprow,oppnum,t,counter=0,num;
 5
         int i,s;
         scanf("%d",&t);
 6
 7
         while(t--)
 8
 9
             s=0;
             scanf("%d",&n);
10
11
             num=1;
12
             opprow=n*n+1;
13
             printf("Case #%d\n",++counter);
             for(row=n;row>=1;row--,opprow=opprow-row)
14
15 •
             {
                  for(i=0;i<s;i++)</pre>
16
                 printf("**");
17
18
                  s++;
                  for(col=1;col<=row;col++)</pre>
19
                      printf("%d0",num++);
20
                  oppnum=opprow;
21
22
                  for(col=1;col<row;col++)</pre>
                    printf("%d0".oppnum++):
23
```

	Input	Expected	Got	
~	3	Case #1	Case #1	~
	3	10203010011012	10203010011012	
	4	**4050809	**4050809	
	5	****607	****607	
		Case #2	Case #2	
		1020304017018019020	1020304017018019020	
		**50607014015016	**50607014015016	
		****809012013	****809012013	
		*****10011	*****10011	
		Case #3	Case #3	
		102030405026027028029030	102030405026027028029030	
		**6070809022023024025	**6070809022023024025	
		****10011012019020021	****10011012019020021	
		*****13014017018	*****13014017018	
		******15016	******15016	

Passed all tests! ✓

Finish review

Quiz navigation







Show one page at a time

Finish review