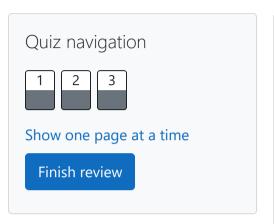
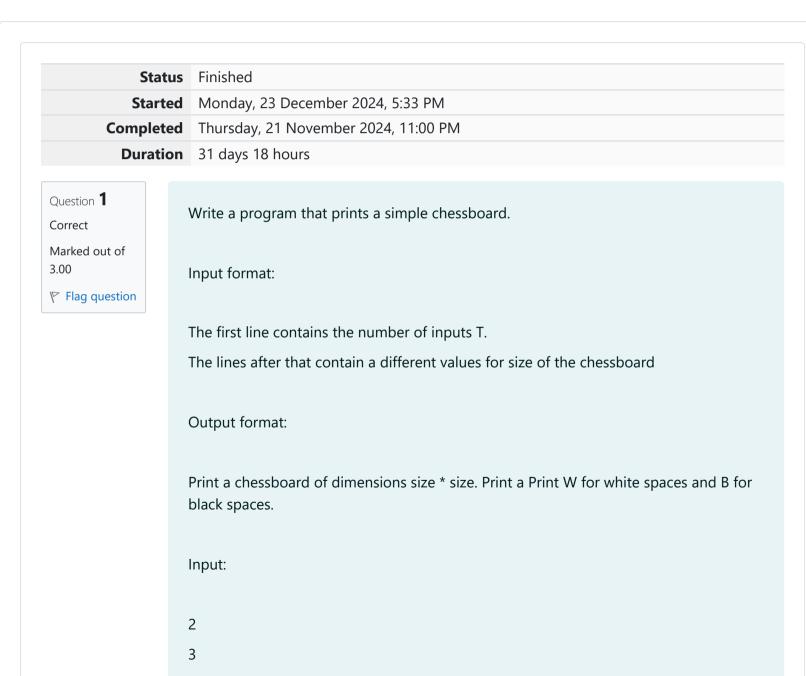
GE23131-Programming Using C-2024





WBW **BWB** WBW **WBWBW BWBWB WBWBW BWBWB WBWBW Answer:** (penalty regime: 0 %) 1 #include<stdio.h> 2 int main() 3 🔻 int t,d,i=0,i1,i2,o; char c; scanf("%d",&t); while(i<t)</pre> 8 , scanf("%d",&d); 9 10 i1=<mark>0</mark>; 11 while(i1<d)</pre> 12 🔻 13 o=**1**; 14 i2=**0**; 15 if(i1%2==0) 16 🔻 17 o=<mark>0</mark>; 18 19 while(i2<d)</pre> 20 🔻

Output:

```
24
                             c='W';
25
                        printf("%c",c);
26
                         i2++;
27
28
29
                    i1+=<mark>1</mark>;
                    printf("\n");
30
31
32
               i=i+1;
33
34
```

	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! ✓

Question **2**

Correct

Marked out of 5.00

▼ Flag question

Let's print a chessboard!

Write a program that takes input:

```
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 %)
   1 #include<stdio.h>
      int main()
```

int t.d.i.i1.i2.o.z:

```
8 🔻
             scanf("%d %c",&d,&s);
 9
             for(i1=0;i1<d;i1++)</pre>
10
11 🔻
                 z=(s=='W')?0:1;
12
                 o=(i1%2==z)?0:1;
13
                 for(i2=0;i2<d;i2++)</pre>
14
15 🔻
                      c=(i2%2==o)?'W':'B';
16
                      printf("%c",c);
17
18
                 printf("\n");
19
20
21
22 }
```

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

Passed all tests! <

Decode the logic and print the Pattern that corresponds to given input.

If N= 3

Marked out of 7.00 Flag question

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

3 3 5 Output Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 *****10011 Case #3 102030405026027028029030 **6070809022023024025 ****10011012019020021 *****13014017018 ******15016

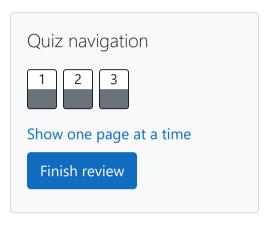
```
|int main()
 3 ▼
    {
         int n,v,p3,c,in,i,i1,i2,t,ti;
 4
         scanf("%d",&t);
 5
         for(ti=0;ti<t;ti++)</pre>
 6
 7 1
 8
             v=<mark>0</mark>;
             scanf("%d",&n);
 9
10
             printf("Case #%d\n",ti+1);
             for(i=0;i<n;i++)</pre>
11
12 🔻
13
                  c=0;
                  if(i>0)
14
15 🔻
16
                       for(i1=0;i1<i;i1++)</pre>
17
                       printf("**");
18
19
                  for(i1=i;i1<n;i1++)</pre>
20
                       if(i>0)
21
22
                       C++;
23
                       printf("%d0",++v);
24
25
                  if(i==0)
26
27
                       p3=v+(v*(v-1))+1;
28
                       in=p3;
29
30
                  in=in-c;
                  p3=in;
31
                  for(i2=i;i2<n;i2++)</pre>
32
33 🔻
                       printf("%d",p3++);
34
35
                       if(i2!=n-1)
                       printf("0");
36
37
38
                  printf("\n");
39
40
         return 0;
41
42 }
```

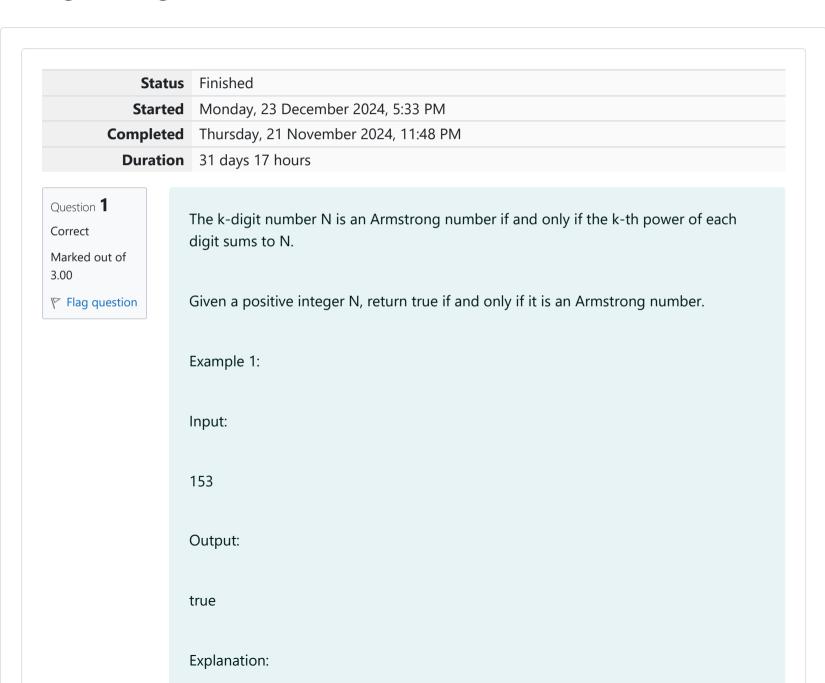
	Input	Expected	Got	
/	3	Case #1	Case #1	~
	3	10203010011012	10203010011012	
	4	**4050809	**4050809	
	5	****607	****607	
		Case #2	Case #2	
		1020304017018019020	1020304017018019020	
		**50607014015016	**50607014015016	
		****809012013	****809012013	
		*****10011	*****10011	
		Case #3	Case #3	
		102030405026027028029030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

Passed all tests! ✓

Finish review

GE23131-Programming Using C-2024





	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:	
	1634	
	Output:	
	true	

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

Passed all tests! ✓

Question **2**

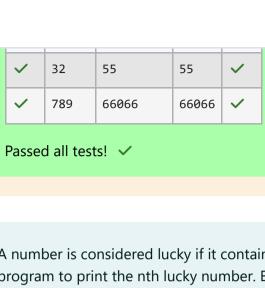
Correct

Marked out of 5.00

▼ Flag question

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
    int main()
 2
 3 🔻
         int rn,n,nt=0,i=0;
 4
 5
         scanf("%d",&n);
         do{
 6 ,
 7
             nt=n;
 8
             rn=<mark>0;</mark>
             while(n!=0)
 9
10
                 rn=rn*10+n%10;
11
12
                 n=n/10;
13
             n=nt+rn;
14
15
             i++;
16
17
         while(rn!=nt || i==1);
         printf("%d",rn);
18
19
         return 0;
20
```



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.

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;
 4
          scanf("%d",&e);
          while(i<e)</pre>
 7 🔻
 8
               nt=n;
               while(nt!=0)
10
11
                    co=0;
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
                   i++:
```

