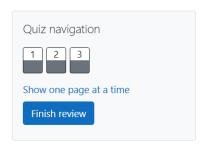
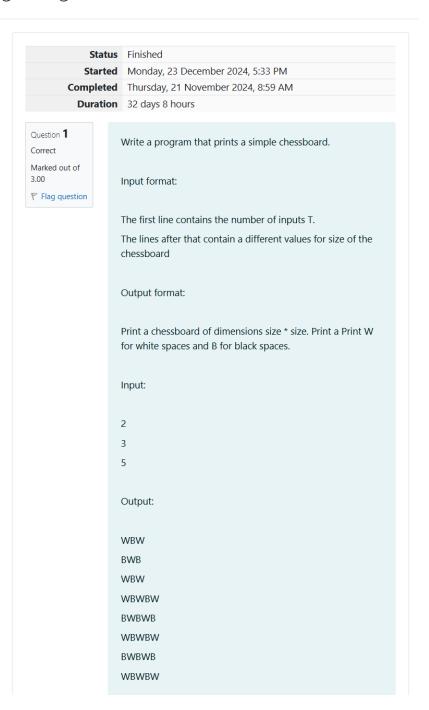
GE23131-Programming Using C-2024





```
Answer: (penalty regime: 0 %)
    1 #include<stdio.h>
       int main()
    3 🔻
    4
            int T,d,i=0,i1,i2,o;
            char c;
scanf("%d",&T);
for(i=0;i<T;i++)</pre>
    5
    6
    7
   8 ,
                 scanf("%d",&d);
    9
   10
                 for(i1=0;i1<d;i1++)</pre>
  11
  12 🔻
                     0=1;
  13
                     i2=0;
   14
  15
                     if(i1%2==0){
                          o=0;
  16
  17
                     for(i2=0;i2<d;i2++)</pre>
   18
  19
   20
                          c='B';
                          if(i2%2==o)
c='W';
   21
   22
                          printf("%c",c);
   23
  24
  25
                     printf("\n");
   26
   27
   28
   29
            return 0;
   30
   31
```

	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

Friag 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 %)

```
1 #include<stdio.h>
           int main()
                  int T,d,o,z;
char c,s;
scanf("%d",&T);
for(int i=0;i<T;i++)</pre>
  4
5
  6
 8 1
                           scanf("%d %c",&d,&s);
for(int i1=0;i1<d;i1++){
    z=(s=='\w')?0:1;
    o=(i1\%2=z)?0:1;
    for(int i2=0;i2<d;i2++){
        c=(i2\%2==0)?'\w':'\B';
        printf("\%c",c);
}</pre>
10 •
11
12
13 ·
14
15
16
                                     printf("\n");
17
18
19
20 21 }
                   return 0;
```



Question 3 Decode the logic and print the Pattern that corresponds to Correct given input. Marked out of 7.00 If N= 3 ▼ Flag question 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 *****10011 Case #3 102030405026027028029030

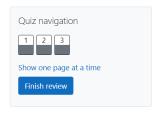
> **6070809022023024025 ****10011012019020021 *****13014017018 *******15016

```
Answer: (penalty regime: 0 %)
       #include<stdio.h>
        int main()
    2
    3 ,
             int n,v,p3,c,i,in,i1,i2,t,ti;
scanf("%d",&t);
for(ti=0;ti<t;ti++){</pre>
    4
    5
    6
                  scanf("%d",&n);
printf("Case #%d\n",ti+1);
    8
    9
   10
                  for(i=0;i<n;i++){</pre>
                      c=0;
   11
                       if(i>0){
   12 🔻
                        for(i1=0;i1<i;i1++){
    printf("**");
   13
   14
   15
   16
                       for(i1=i;i1<n;i1++){
   17 ,
                        if(i>0){
   18 🔻
   19
                         C++;
   20
                        printf("%d0",++v);
   21
   22
                       if(i==0){
   23
                       p3=v+(v*(v-1))+1;
   24
   25
                       in=p3;
   26
   27
                       in=in-c;
   28
                       p3=in;
                      for(i2=i;i2<n;i2++){
printf("%d",p3++);
   29
   30
                       if(i2!=n-1)
   31
                      printf("0");
   32
   33
   34
                  printf("\n");
   35
   36
   37
             return 0;
   38 }
```

	Input	Expected	Got
~	3	Case #1	Case #1
	3	10203010011012	10203010011012
	4	**4050809	**4050809
	5	****607	****607
		Case #2	Case #2
		1020304017018019020	10203040170180
		**50607014015016	**506070140150
		****809012013	****809012013
		*****10011	*****10011
		Case #3	Case #3
		102030405026027028029030	10203040502602
		**6070809022023024025	**607080902202
		****10011012019020021	****1001101201
		*****13014017018	*****13014017
		******15016	*******15016
	d all test		

Finish review

GE23131-Programming Using C-2024





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

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

Passed all tests! <

Question **2**Correct
Marked out of 5.00

Friag 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 %)

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

	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 34 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 %)

```
| Input | Expected | Got |
| ✓ | 34 | | 33344 | | ✓ |
| Passed all tests! ✓ |
```

Finish review