

1. Write a multifunction program to print the following patterns where number of rows is user input and must be read in main function. There should be separate function for each of the following patterns and note that, you cannot pass any data through parameters to those functions.

(a) 4444444  
33333  
222  
1

(b) 7654321  
54321  
321  
1

(c) \* \* \* \* \*  
\* \* \*  
\*  
\* \* \*  
\* \* \* \* \*

The screenshot shows a C program in a code editor with tabs for 'problem2.c', 'problem3.c', 'exercise1.c', and 'exercise2'. The code defines three functions: patternA, patternB, and patternC, each taking a line number as input. The main function prompts the user for a line number and calls these functions. The output window shows the program's execution for line number 4, displaying Pattern A (4444444), Pattern B (7654321), and Pattern C (\*\*\*\*\*). The process returns 0 (0x0) and prompts the user to press any key to continue.

```
#include<stdio.h>//2204044
int line;//2204044
int patternA();//2204044
int patternB();//2204044
int patternC();//2204044
int main(){
    printf("Enter line number :");//2204044
    scanf("%d",&line);//2204044
    patternA();//2204044
    patternB();//2204044
    patternC();//2204044
    return 0;//2204044
}
int patternA(){
    int i,j=line;
    printf("Pattern A:\n");//2204044
    while(j){
        for(int i=j*2-1;i>0;i--){
            printf("%d",j);//2204044
        }
        printf("\n");//2204044
        j--;//2204044
    }
}
int patternB(){
    int i,j=line;
    printf("\nPattern B:\n");//2204044
    while(j){
        for(int i=j*2-1;i>0;i--){
            printf("%d",i);//2204044
        }
        printf("\n");//2204044
        j--;
    }
}
```

Enter line number :4  
Pattern A:  
4444444  
33333  
222  
1  
  
Pattern B:  
7654321  
54321  
321  
1  
  
Pattern C:  
\*\*\*\*\*  
\*\*\*  
\*  
\*\*\*  
\*\*\*\*\*  
  
Process returned 0 (0x0)    exec  
Press any key to continue.

```

int patternC() {
    int i, j=line+1; //2204044
    printf("\nPattern C:\n"); //2204044
    while(j>1) {
        for(int i=j; i>0; i--) {
            printf("*"); //2204044
        }
        printf("\n"); //2204044
        j-=2; //2204044
    }
    while(j<=line+1) {
        for(int i=j; i>0; i--) {
            printf("*"); //2204044
        }
        printf("\n"); //2204044
        j+=2; //2204044
    }
}

```

2. Write a function to calculate the factorial value of any integer entered through the keyboard.

The screenshot shows a C program in a code editor with tabs for 'problem2.c', 'problem3.c', 'exercise1.c', and 'exercise2.c'. The code defines a recursive function 'fact' to calculate the factorial of a number 'n'. The 'main' function prompts the user to 'Enter an Integer :', reads the input, and prints the result using 'fact(num)'. The output window shows the program running, with the input '8' and the output 'Factorial :40320'. It also displays 'Process returned 0 (0x0)' and 'Press any key to continue.'

```

#include<stdio.h> //2204044
int fact(int n); //2204044
int main() {
    int num; //2204044
    printf("Enter an Integer :"); //2204044
    scanf("%d", &num); //2204044
    printf("Factorial :%d", fact(num)); //2204044
    return 0; //2204044
}
int fact(int n) {
    if(n>0) {
        return n*fact(n-1); //2204044
    }
    return 1; //2204044
}

```

Enter an Integer :8  
 Factorial :40320  
 Process returned 0 (0x0) exec  
 Press any key to continue.

3. A prime integer is entered through the keyboard. Write a function to obtain the prime factors of this number. For example, prime factors of 24 are 2, 2, 2 and 3 whereas prime factor of 35 are 5 and 7.

```
problem2.c X problem3.c X exercise1.c X exercise2
#include <stdio.h>//2204044
void primeFactor(int n);//2204044
int main(){
    int num;//2204044
    printf("Enter an Integer :");//2204044
    scanf("%d",&num);//2204044
    printf("Prime Factor :");//2204044
    primeFactor(num);//2204044
    return 0;//2204044
}
void primeFactor(int n){
    if(n>=2){
        for(int i=2;i<=n;i++){
            if(n%i==0){
                printf("%d ",i);//2204044
                primeFactor(n/i);//2204044
                break;//2204044
            }
        }
    }
}
```

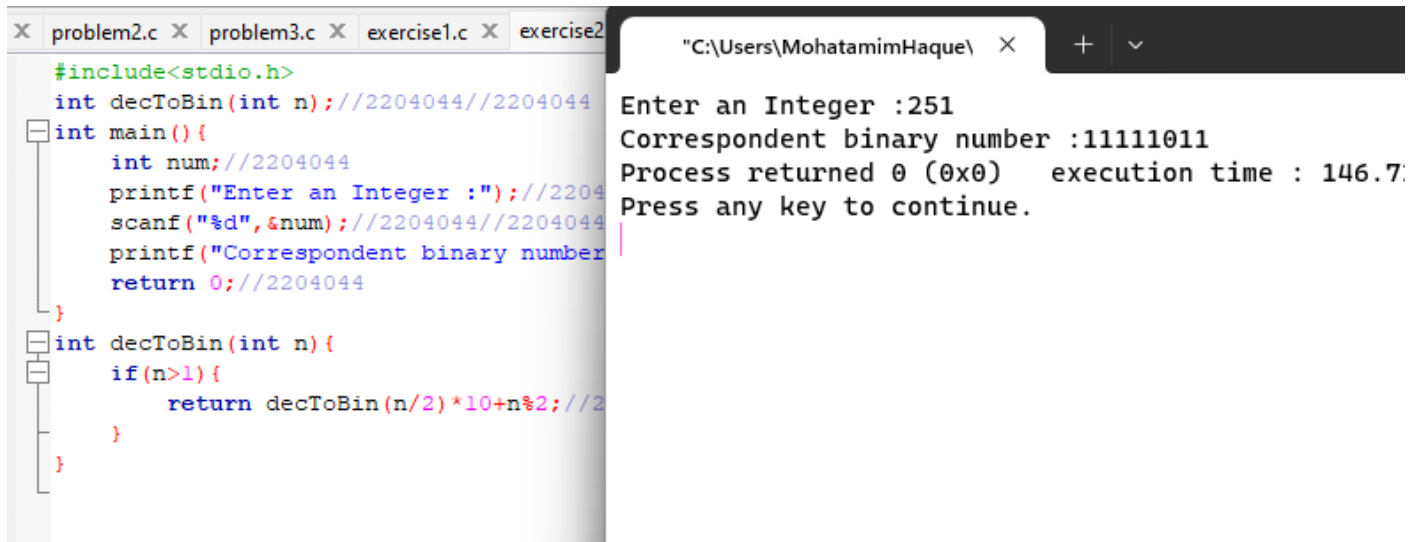
"C:\Users\MohatamimHaque\ X +  
Enter an Integer :48  
Prime Factor :2 2 2 2 3  
Process returned 0 (0x0) execution time : 0.000 s  
Press any key to continue.

1. The series 0, 1, 1, 2, 3, 5, 8, 13, ... is called the Fibonacci series. Here,  $term_n = term_{n-1} + term_{n-2}$ , for  $n > 1$ ,  $term_0 = 0$ ,  $term_1 = 1$ . Write a program that finds the sum of first  $n$  terms of the series using recursion.

```
problem2.c X problem3.c X exercise1.c X exercise2
#include <stdio.h>//2204044
int fibonacci(int n);//2204044
int main(){
    int num;//2204044
    printf("Enter Terms :");//2204044
    scanf("%d",&num);//2204044
    printf("First %d terms fibonacci Sum :");//2204044
    return 0;//2204044
}
int fibonacci(int n){
    static int a=0,b=1;//2204044
    if(n>0){
        int c=a;//2204044
        b=b+a;//2204044
        a=b-a;//2204044
        return c+fibonacci(n-1);//2204044
    }
    else return 0;//2204044
}
```

"C:\Users\MohatamimHaque\ X +  
Enter Terms :9  
First 9 terms fibonacci Sum :54  
Process returned 0 (0x0) execution time : 0.000 s  
Press any key to continue.

2. Convert a decimal number into correspondent binary number using recursion where decimal number is input from user.



The image shows a code editor with four tabs: 'problem2.c', 'problem3.c', 'exercise1.c', and 'exercise2'. The 'exercise2' tab is active, displaying a C program that converts a decimal number to binary using recursion. The program includes `<stdio.h>`, defines a recursive function `decToBin`, and a `main` function that prompts the user for an integer, reads it, and prints the corresponding binary number. The output window on the right shows the program's execution: it prompts 'Enter an Integer :', the user enters '251', and the program outputs 'Correspondent binary number :11111011'. It also displays 'Process returned 0 (0x0)' and 'execution time : 146.7'. A prompt 'Press any key to continue.' is visible at the bottom of the output window.

```
#include<stdio.h>
int decToBin(int n); //2204044//2204044
int main(){
    int num; //2204044
    printf("Enter an Integer :"); //2204
    scanf("%d",&num); //2204044//2204044
    printf("Correspondent binary number
    return 0; //2204044
}

int decToBin(int n){
    if(n>1){
        return decToBin(n/2)*10+n%2; //2
    }
}
```

Enter an Integer :251  
Correspondent binary number :11111011  
Process returned 0 (0x0) execution time : 146.7  
Press any key to continue.