

University of Rajshahi
Department of Computer Science and Engineering
B.Sc. Engineering **Part-I, Odd Semester Examination 2018(2016-2017)**
Course Code: **CSE-1121** Course Title: **Computer Programming with C**
Time: **03 Hours** Full Marks: **52.5**

Section **A**
Answer any **THREE** questions.

1. (a) What is run-time error? Give some examples of run-time error? 2.25
(b) If $i=4$, $j=2.5$ and `int answer`, what will be the output of `answer=(i+j)%2`? 1.50

1 a) Answer: A runtime error in a program is an error that occurs while the program is running after being successfully compiled. Runtime errors are commonly called referred to as “bugs” and are often found during the debugging process before the software is released.

```
#include<stdio.h>
int main(){
    int a=5,b=0;
    printf("%d",(a/b));
    return 0;
}
```

Dividing with 0 gives infinite solution, so this is run-time error.

Another example is when the index of the array is assigned with a negative index it leads to invalid memory access during runtime error.

b) Answer: Given $i=4$ and $j=2.5$. Here, i is int type and j is float type.

If we add i and j we get 6.5. And if we store the value in an integer type variable then we will get only 6. Suppose we have `int sum= i+j`; then `sum=6`.

After that `answer=(i+j)%2 = (6)%2 = 0`

So we get 0 as answer.

(c) List the syntax error (if any) of each line of the following code.

5.00

```
struct Books {
    char title[50];
    char author[50];
    char subject[100];
    int book_id;
};
double printBook( struct Books book ) {
    printf( "Book title : %f\n", book.title);
    printf( "Book author : %s\n", book.author);
    printf( "Book subject : %s\n", book.subject);
    printf( "Book book_id : %s\n", book.book_id);
}

int main( ) {

    struct Books Book1;
    struct Books Book2;

    strcpy( Book1.title, "C Programming");
    strcpy( Book1.author, "Prata");
    strcpy( Book1.subject, "C Programming Tutorial");
    Book1.book_id = "6495407";

    strcpy( "Telecom Billing", Book2);
    strcpy( Book2.author, "Goodman");
    strcpy( Book2.subject, "Telecom Billing Tutorial");
    Book2.book_id = "6495700";

    printBook( Books Book1 );
    printBook( Book2 );

    return 1;
}
```

1 c) Answer:

The errors of the given code are highlighted below:

```
1  #include<stdio.h>
2  #include<string.h>
3
4  struct Books{
5      char title[50];
6      char author[50];
7      char subject[100];
8      int book_id;
9  };
10
11 void printBook(struct Books book){
12     printf("Book title: %s\n",book.title);
13     printf("Book author: %s\n",book.author);
14     printf("Book subject: %s\n",book.subject);
15     printf("Book book_id: %d\n", book.book_id);
16
17 }
18
19 int main(){
20
21     struct Books Book1;
22     struct Books Book2;
23
24     strcpy( Book1.title, "C Programming");
25     strcpy(Book1.author, "Prata");
26     strcpy(Book1.subject, "C programming Tutorial");
27     Book1.book_id = 6495407;
28
29     strcpy( Book2.title, "Telecom Billing");
30     strcpy(Book2.author, "Goodman");
31     strcpy(Book2.subject, "Telecom Billing tutorial");
32     Book2.book_id=6595700;
33
34     printBook(Book1);
35     printBook(Book2);
36
37     return 1;
38 }
```

2. (a) What will be the output of the following code?

3.75

```
#include <stdio.h>
```

```
int a = 20;
```

```
int e;
```

```
int main () {
```

```
    int a = 10;
```

```
    int b = 20;
```

```
    int c = 0;
```

```
    int d = 0;
```

```
    printf("value of a in main() = %d\n", a);
```

```
    c = sum( a, b);
```

```
    d = sub( a, b);
```

```
    printf("value of c in main() = %d\n", c);
```

```
    printf("value of d in main() = %d\n", d);
```

```
    printf("value of e in main() = %d\n", e);
```

```
    return 0;
```

```
}
```

```
int sum(int a, int b) {
```

```
    printf("value of a in sum() = %d\n", a);
```

```
    printf("value of b in sum() = %d\n", b);
```

```
    return a + b;
```

```
}
```

```
int sub(int a, int b) {
```

```
    int c=40;
```

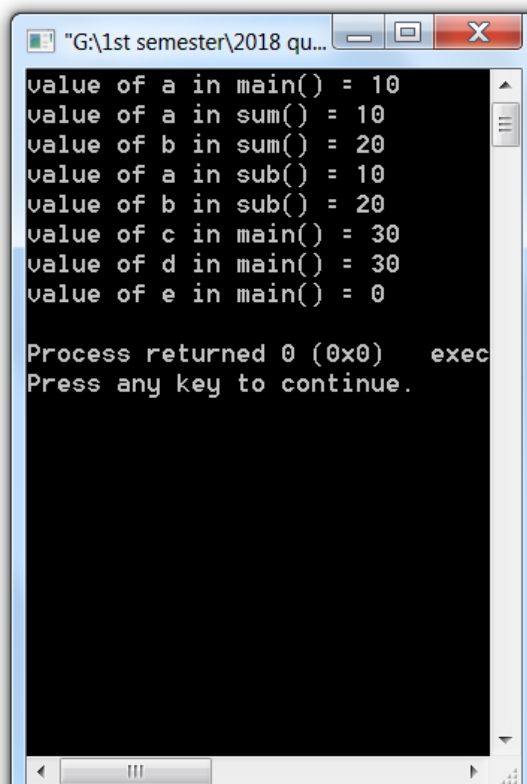
```
    printf("value of a in sub() = %d\n", a);
```

```
    printf("value of b in sub() = %d\n", b);
```

```
    return a-b + c;
```

```
}
```

```
1  #include<stdio.h>
2
3  int a=20;
4  int e;
5
6  int main(){
7      int a=10;
8      int b=20;
9
10     int c=0;
11     int d=0;
12
13     printf("value of a in main() = %d\n",a);
14     c = sum(a,b);
15     d= sub(a,b);
16     printf("value of c in main() = %d\n",c);
17     printf("value of d in main() = %d\n",d);
18     printf("value of e in main() = %d\n",e);
19     return 0;
20 }
21
22 int sum(int a, int b){
23     printf("value of a in sum() = %d\n",a);
24     printf("value of b in sum() = %d\n",b);
25     return a+b;
26 }
27
28 int sub(int a, int b){
29     int c=40;
30     printf("value of a in sub() = %d\n",a);
31     printf("value of b in sub() = %d\n",b);
32     return a-b+c;
33 }
```



```
"G:\1st semester\2018 qu...
value of a in main() = 10
value of a in sum() = 10
value of b in sum() = 20
value of a in sub() = 10
value of b in sub() = 20
value of c in main() = 30
value of d in main() = 30
value of e in main() = 0

Process returned 0 (0x0)   exec
Press any key to continue.
```


(b) What will be the output of the following code? Is there any better or easier way to produce the same output? If any, write the program? 5.00

```
int main()
{
    int v[4][4];
    int n=3;
    for(int i = 1;i <=n;++i)
    for(int j = 1;j <=n;++j)
        v[i-1][j-1]=(i/j)*(j/i);

    for(int i = 1;i <=n;++i)
    {
        for(int j = 1;j <=n;++j)
        {
            printf("%d",v[i-1][j-1]);
        }
        printf("\n");
    }

    return 0;
}
```

The output of this code:

100

010

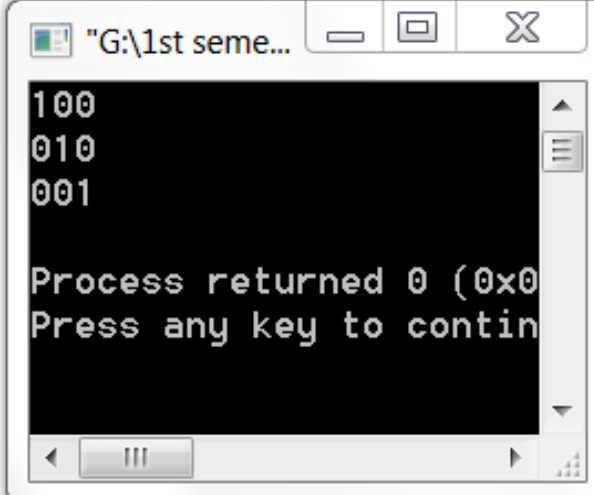
001

Another way to write the code to produce same output:

```

1  #include<stdio.h>
2
3  int main(){
4
5      int value=4;
6
7      for(int i=3;i>=1;i--){
8          printf("%03d\n",decimalToBinary(value));
9          value = value>>1;
10     }
11
12     return 0;
13 }
14 int decimalToBinary(int n){
15     long binarynum = 0;
16     int remainder, temp = 1;
17
18     while (n!=0)
19     {
20         remainder = n%2;
21         n = n / 2;
22         binarynum = binarynum + remainder*temp;
23         temp = temp * 10;
24     }
25     return binarynum;
26 }

```



```

#include<stdio.h>
int decimalToBinary(int n);
int main(){
    int value=4;
    for(int i=3;i>=1;i--){

printf("%03d\n",decimalToBinary(value));
        value = value>>1;
    }
    return 0;
}

```

```

int decimalToBinary(int n){
    long binarynum = 0;
    int remainder, temp = 1;

    while (n!=0)
    {
        remainder = n%2;
        n = n / 2;
        binarynum = binarynum +
remainder*temp;
        temp = temp * 10;
    }
    return binarynum;
}

```

3.	(a)	How do logical errors differ from syntactic and execution errors?	2.00
	(b)	What are the purposes of using continue and break statement? Give example	3.00
	(c)	What is the purpose of using do-while statement? How does it differ from the while statement? Explain with example.	3.75

3 a) Answer:

Logic error

An error in the logic of the code such as using < instead of > or AND instead of OR. The program will execute the code but will produce **unexpected results**. Logic errors are usually resolved by carrying out a dry run, using a trace table or setting breakpoints to help identify the section of code that contains the logic error.

Where as the syntactic and execution errors are:

Syntax error

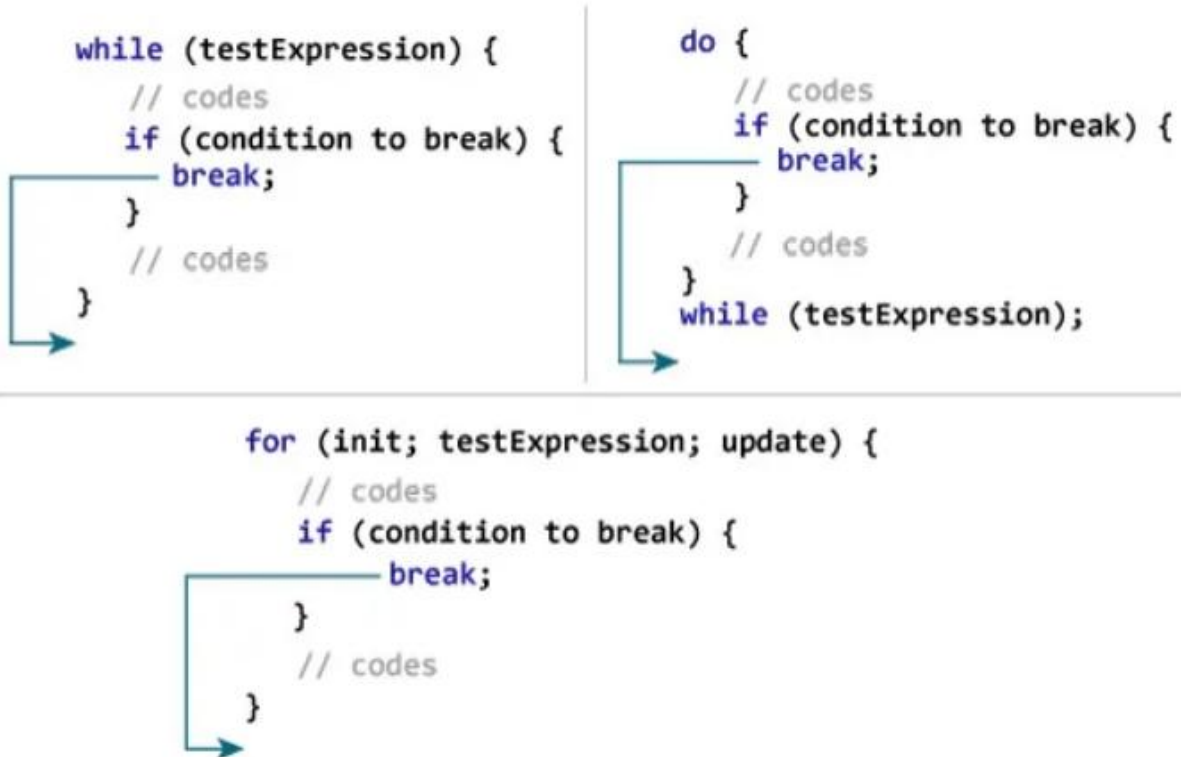
This is an error in the **spelling or grammar used when coding**. If the code is **missing a letter, character or forgetting to include inverted commas/speech marks are common examples of syntax errors**. A syntax error will be identified by an interpreter as it will be unable to convert the source code into machine code.

Execution error

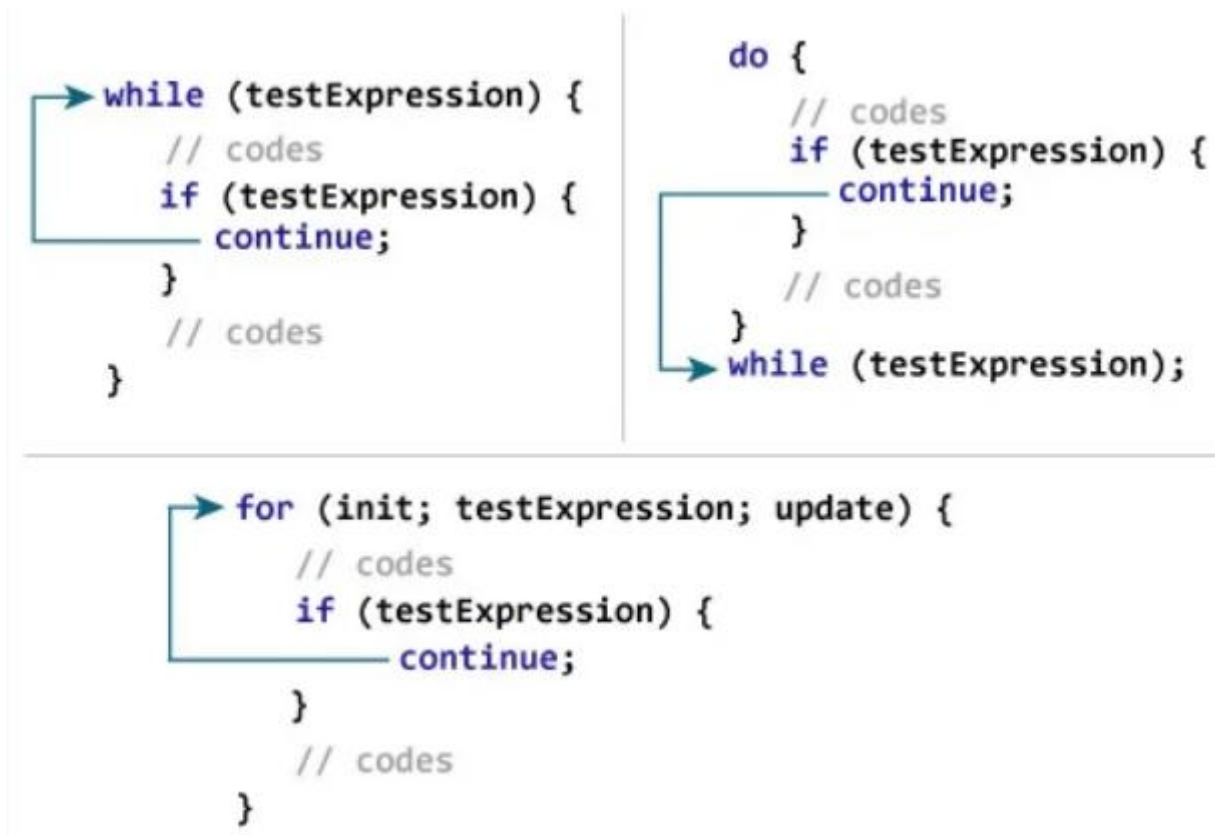
Sometimes called **a runtime error**, execution errors only become evident during run time. An execution error occurs when a **program is asked to do something that it cannot, result in a 'crash'**. The widely used example of a run time error is asking a program to divide by 0. The code contains no syntax or logic errors but when it runs it can't perform the task that it has been programmed to carry out. Another example is where an attempt is made to access an item in an array which does not exist eg item 11 in an array with only ten elements. File handling can also result in execution errors, most often when an attempt is made to write to a file that does not exist. So, this is how they differ from each other.

3.	(a)	How do logical errors differ from syntactic and execution errors?	2.00
	(b)	What are the purposes of using continue and break statement? Give example	3.00

3 b) Answer: The break statement ends the loop immediately when it is encountered. The break statement is almost always used with if...else statement inside the loop.



The continue statement skips the current iteration of the loop and continues with the next iteration.



3 c) Answer: The do-while loop executes the body of the loop at least once even if the condition is false at the first attempt.

Key Differences Between while and do-while Loop

1. The while loop checks the condition at the starting of the loop and if the condition is satisfied statement inside the loop, is executed. But in the do-while loop, the condition is checked after the execution of all statements in the body of the loop.
2. If the condition in a while loop is false, not a single statement inside the loop is executed. In contrast, if the condition in 'do-while' loop is false, then also the body of the loop is executed at least once then the condition is tested.
3. The while loop is also known as the entry-controlled and pre-checking loop because in this loop, the body of the loop is executed prior to checking the condition. Conversely, the alternate name for the do-while loop is the exit-controlled and post-checking loop, the reason behind this is that the checking of the loop condition is followed by the execution of the body of the loop.
4. The syntax of a do-while loop includes a semi-colon to terminate the loop. On the contrary, there is no use of the semi-colon in the while loop syntax.

```
while ( condition) {  
statements; //body of loop  
}
```

```
do{  
statements; // body of loop.  
} while( condition );
```

4. (a) What will be the output of the following code?

```
void operation(int *num1, int *num2)
{
    int tempnum;

    tempnum = *num1;
    *num1 = *num2;
    *num2 = tempnum;
}

int main( )
{
    int v1 = 11, v2 = 77 ;
    printf("Before operation:");
    printf("\nValue of v1 is: %d", v1);
    printf("\nValue of v2 is: %d", v2);

    operation( &v1, &v2 );

    printf("\nAfter operation:");
    printf("\nValue of v1 is: %d", v1);
    printf("\nValue of v2 is: %d", v2);
}
```

Output:

```
Before operation:
Value of v1 is: 11
Value of v2 is: 77
After operation:
Value of v1 is: 77
Value of v2 is: 11
Process returned 0 (0x
Press any key to conti
```

(b) A 4x4 matrix is given in a file named in.txt.

in.txt:

```
10    15    20    25
5     10    20    15
15    25    10    5
20    5     15    10
```

Read the matrix from the in.txt file, transform all the decimal numbers to its equivalent binary numbers and write into out.txt file.

out.txt:

```
1010  1111  10100 11001
101   1010  10100 1111
1111  11001 1010  101
10100 101   1111  1010
```

4 b) Solution:

```
#include<stdio.h>
```

```
int main(){
```

```
    int num[100];
```

```
    FILE *fptr, *fptr2;
```

```
    if((fptr = fopen("in.txt","r"))==NULL){
```

```
        printf("Error opening file");
```

```
        exit(1);
```

```
    }
```

```
    int i;
```

```
    fptr2 = fopen("out.txt","w");
```

```
for(i=0;i<4;i++){  
    for(int j=0;j<4;j++){  
        fscanf(fptr,"%d", &num[j]);  
  
        fprintf(fptr2, "%-5d ",decimalToBinary(num[j]));  
    }  
    fprintf(fptr2,"\n");  
}  
  
fclose(fptr2);  
return 0;  
}
```

```
int decimalToBinary(int n){  
    long binarynum = 0;  
    int remainder, temp = 1;  
  
    while (n!=0)  
    {  
        remainder = n%2;
```


$n = n / 2;$

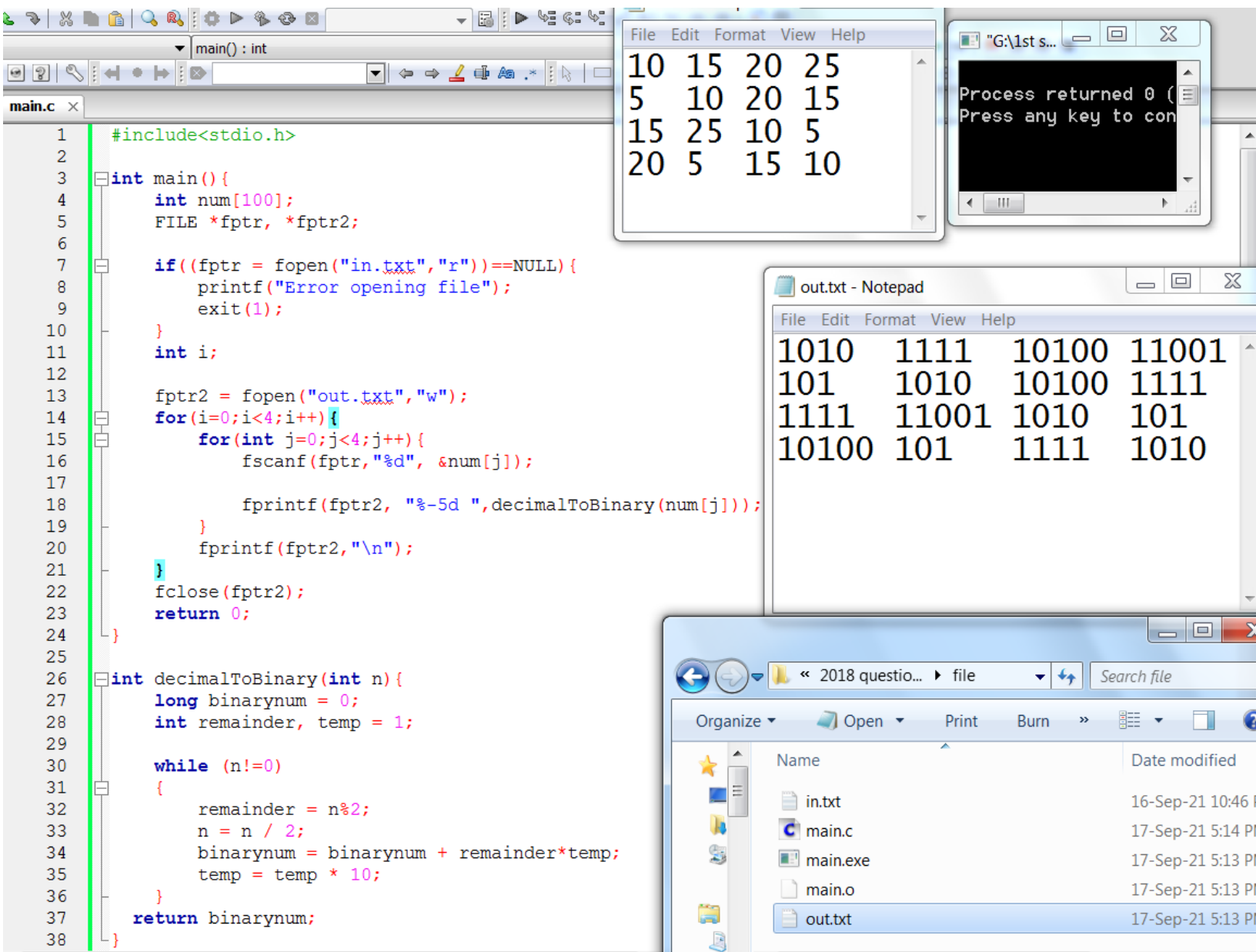
$\text{binarynum} = \text{binarynum} + \text{remainder} * \text{temp};$

$\text{temp} = \text{temp} * 10;$

}

return binarynum;

}



Any query: 18115@imperial.edu.bd