University of Rajshahi
Department of Computer Science and Engineering
B.Sc. (Engg.) Part-1 Odd Semester Examination 2015
Course: CSE 1121 (Computer Programming with C)
Full Marks: 52.5 Time: 3 Hours
Answer Six (06) Questions taking any Three (03) from each section in separate answer script

Section A

1 (a) What is run-time error? Give an example. 2

1 a) Answer: Runtime Error: A runtime error in a program is an error that occurs while the program is running after being successfully compiled.

There are a variety of runtime errors that occur such as **logical errors**, **Input/Output errors**, **undefined object errors**, **division by zero errors**, and many more.

```
int main(){
    int a = 5;
  // Division by Zero
   Printf("%d", a / 0);
   return 0;
}
```

The program will create a run time error.
Because the output is not possible.

(b) What are local and global variables? 2

Topics	**Local Variables**.	**Global Variables**.
Definition	Those variables which are defined within some function and are accessible to that function only are called **Local Variables**.	Those variables which are defined outside of function block and are accessible to entire program are known as **Global Variables**.
Scope	Scope is local to that block or function where they are defined.	Scope is global i.e. they can be used anywhere in the program.
Default value	Default value is unpredictable (garbage).	Default value is Zero (0).

Example	**#include**<stdio.h>
	void main()
	{
	int x=23, y=4;
	printf("x = %d and y=%d",x,y);
	}
	Here x and y are local variables
		#include<conio.h>
		int a=10,b;
		void main()
		{
		printf("a = %d and b=%d",a,b);
		}
		Here a and b are global variables.

2015 1(c) same as 2016 2(b)

2.(a) What are the uses of 'break' statements? Give an example 3

The break is a keyword in C which is used to bring the program control out of the loop. The break statement is used inside loops or switch statement. The break statement breaks the loop one by one, i.e., in the case of nested loops, it breaks the inner loop first and then proceeds to outer loops. The break statement in C can be used in the following two scenarios:

1. With switch case
2. With loops

```
#include<stdio.h>
#include<stdlib.h>
void main ()  {
   int i;
   for(i = 0; i<10; i++)  {
      printf("%d ",i);
      if(i == 5)
      break;
   }
   printf("came outside of loop i = %d",i);
}
```

Output

0 1 2 3 4 5 came outside of loop i = 5

b) Define escape sequence. List any 5(five) escape sequences used in C programming. 2

2b) Answer: An escape sequence in C language is a sequence of characters that doesn't represent itself when used inside string literal or character.

It is composed of two or more characters starting with backslash \. For example: \n represents new line.

Escape Sequence	Meaning
\a	Alarm or Beep
\b	Backspace
\v	Vertical Tab
\\	Backslash
\'	Single Quote

c) Distinguish between unary and binary operator with example. 2

Unary Operators	Binary Operators
(i) The operators which act upon a single operand are called unary operators.	(i) The operators which require two operands for their action are called binary operators.
(ii) They are pre-increment and post increment (+ +).	(ii) They are mathematical operators and relational operators.

c) Distinguish between unary and binary operator with example.

d) Find the output of the following code. 1.75

```
void main() {
    int i = 10, j = 20 ;
    float a, b, c;
    a = i / j;
    b = 1.0 * i / j ;
    c = i / j * 1.0;
    printf("%f %f %f ", a, b, c);  }
```

d.c

```
1  #include<stdio.h>
2
3  void main(){
4      int i=10,j=20;
5      float a,b,c;
6      a = i / j;
7      b = 1.0 * i / j;
8      c = i / j *1.0;
9      printf("%f %f %f ",a,b,c);
10 }
```

"G:\1st semester\2016 q...

```
0.000000 0.500000 0.000000
Process returned 27 (0x1B)
Press any key to continue.
```

2.(a) What are the uses of 'break' statements? Give an example 3

Same as 2018 3b

(b) What will be the output of the following statements? 3

```
printf("%5d\n",123);
printf("%-5d\n",123);
printf("%05d\n",15);
printf("%3.2f\n",3.14159);
printf("%x\n",255);
printf("%o\n",255);
```

rt here | 2b.c

```
1  #include<stdio.h>
2
3  int main(){
4      printf("%5d\n",123);
5      printf("%-5d\n",123);
6      printf("%05d\n",15);
7      printf("%3.2f\n",3.14159);
8      printf("%x\n",255);
9      printf("%o\n",255);
10     return 0;
11 }
```

"G:\1st semester\2015 question solve

```
  123
123
00015
3.14
ff
377

Process returned 0 (0x0)
```

printf(%d\n",255);

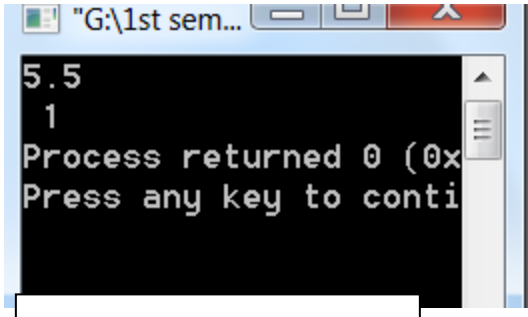
(c) Write a function named *' int floatInteger(float n)'* to decide whether a number, n, is a floating point or pure integer. Your function will return '1', if n is a floating point number otherwise '0'. 2.75

2c.c

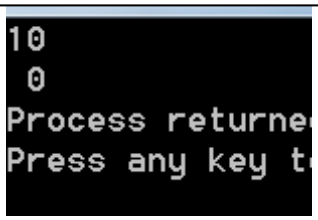
```c
#include<stdio.h>
int floatInteger(float n);
int main(){
    float value;
    scanf("%f",&value);
    printf(" %d ", floatInteger(value));
    return 0;
}

int floatInteger(float n){
    int t = (int)n;
    if(n == t){
        return 0;
    }else{
        return 1;
    }
}
```

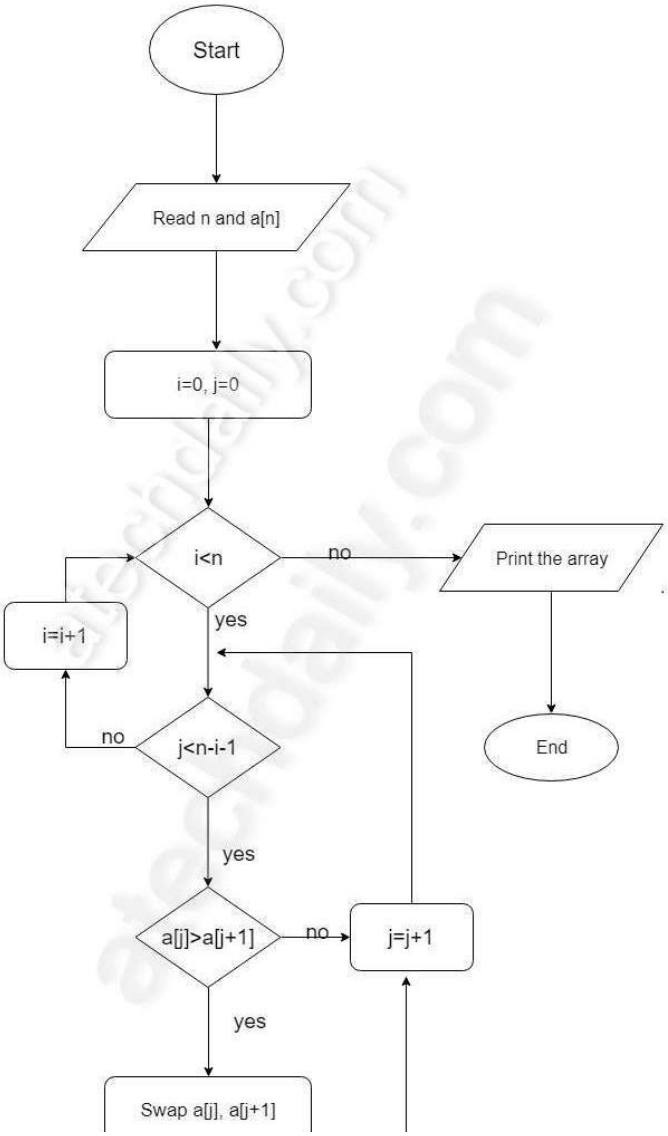
Floating point number:



Integer number:



3.(a) Suppose, you are given an array of 'n' integers. You are asked to develop a program to sort that array in ascending order using at most one extra variable. Draw a flowchart to solve the problem. 4



(b) What will be output of the following code? 4.75

```
int x[10] = {1, 4, 3, 6, 8, 2, 9, 0, 5, 7};
int i,j,k,tmp,big,p;
main() {
 for (i=1;i<=5;i++)
 {
 big=x[i];
  for(j=i; j<=5; j++)
     {if(x[j]> big) p=j+1;}
     tmp=x[p];     x[p]=x[i];       x[i]=tmp;
}
for (k=1;k<8;k++) printf ("%d -th  %d\n",k,x[k]);
}
```

3b.c

```c
#include<stdio.h>
#include<conio.h>
int x[10]={1,4,3,6,8,2,9,0,5,7};
int i,j,k,tmp,big,p;

int main(){
    for(i=1;i<=5;i++){
        big = x[i];
        for(j=i;j<=5;j++){
            if(x[j]>big){
                p = j+1;
            }
        }
        tmp= x[p];
        x[p]=x[i];
        x[i]=tmp;
    }
    for(k=1;k<8;k++){
        printf("%d -th %d\n",k,x[k]);
    }
    return 0;
}
```

Output:

"G:\1st semester\2015 question solv

```
1 -th 2
2 -th 9
3 -th 4
4 -th 6
5 -th 8
6 -th 3
7 -th 0

Process returned 0 (0x0)
```

(c) Why do we need to use '*fclose(fp)*'? 1

3 c) Answer: And when we need to close the file after processing file it needs to close that file, and for that we use that file handling function fclose().