CS DEPT
MILITARY COLLEGE OF SIGNALS, NUST
FUNDAMENTALS OF PROGRAMMING (FOP)
BESE-16B

Exam:	Final	**Instructor:**	Dr. Faisal Bashir
Type of Paper:	Regular	**Total Marks:**	50
Semester:	Fall	**Time Allowed:**	2 hrs and 30 mins

Instructions:

1. Attempt all questions
2. Write your Index # on question paper and answer book.
3. Please write neatly and number questions and subparts carefully.
4. If a question is unclear, state your assumptions and answer the problem based on your assumptions.
5. Understanding the question is also a part of the examination.

Q1. Write the output of the following code snippets. In case of syntax error(s) indicate what is wrong with the code. [2 x 5 = 10]

a.

```
int i = 5 , j;
int *p , *q;
p = &i;
q = &j;
j = 5;
cout<<*p<<endl<<*q;
```

b.

```
char arr[10];
arr = "Computer";
char  arr1[7]= "Network";
cout<<arr <<endl<<arr1;
```

c.

```
int i = 3;
int *j;
int **k;
j=&i;
k=&j;
cout<<k<<endl<<*k<<endl<<**k;
```

d.

```
int x=1;
if(x--)
   printf("FOP is ");
   printf("EASY");
else
   printf("FOP is DIFFICULT");
```

e.

```
char *str="ONE";
str++;
switch(str){
     case "ONE":printf("FIRST CASE");
          break;
     case "NE": printf("SECOND CASE");
          break;
     case "N":  printf("Third CASE");
          break;
     case "E":  printf("LAST CASE");
}
```

Q2. Re-write the following code snippet after removing all the unnecessary tests from the nested conditional statements. [2]

```
float income;
cout << "Enter your monthly income: ";
cin >> income;
if (income < 0.0)
   cout << "Find some Job." << endl;
else if (income >= 0.0 && income < 5000.00)
   cout << "Hard to Earn." << endl;
else if (income >= 5000.00 && income < 15000.00)
   cout << "Looking for a decent job." << endl;
else if (income >= 15000.00)
   cout << "This is not enough too." << endl;
```

Q3. Draw a flow chart that computes and prints the first ten prime numbers from the Fibonacci series. [4]

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, …..

Q4. Consider the following function of binary search which returns the index (position) of the searched item if found else returns -1. [3+3]

```
int binary_search(int data[],// input: array
                  int size, // input: array size
                  int value  // input: value to find
                 )
{
    int lower, middle, upper;// indexes to the array
    lower = 0;
    upper = size - 1;

    while (true) {
       for (int i = lower; i <= upper; i++)
           cout << data[i]<< "-";
       cout << endl;

        middle = (lower + upper) / 2;
        if (data[middle] == value)
            return middle;
        else if (lower >= upper)
            return -1;

        else if (value < data[middle])
            upper = middle - 1;
        else
            lower = middle + 1;
   }
 }
```

Assuming `int list[]={1,2,3,4,5,6,7,8};`

What are the outputs of the following function calls?

a.

```
cout << binary_search(list, 8, 7) << endl;
```

b.

```
cout << binary_search(list, 8, 0) << endl;
```

Q5. Find the output of the following code. **[5]**

```
int f(int a[][3], int b[][3], int c[][3])
{
     int i, j, k;
     k = 0;
     for( i=0; i<3; i++) {
     for( j=0; j<3; j++) {
     c[i][j] = b[2-i][j] * a[i][2-j];
     k += c[i][j];
     }
     }
     return k;
}

void main()
{
     int a[3][3] = { {1,4,7},{2,5,8},{3,6,9}};
     int b[3][3] = { {1,2,3},{4,5,6},{7,8,9}};
     int c[3][3];
     int d;
     d = f(a,b,c);
     cout << "Return value of f() : "<<d<< endl;
}
```

Q6. Find the output of the following code. **[5]**

```
#define SIZE 10
void f(int a[])
{
     int i, j, k;
     for( i=1; i < SIZE; i++) {
     j = a[i];
     for( k=i-1; k>=0 && a[k]<j; k--)
     a[k+1] = a[k];
     a[k+1] = j;
     }
}
void main()
{    int a[SIZE] = {2,9,5,4,8,1,6,10,3,7};
     int *b, i;
     b = a;
     f(a);
     for( i=0; i<SIZE; i++)
     cout << b[i] << " ";
}
```

Q7. Write prototype and the definition of a function that finds the occurrence of one string in another string. [1+6]

```
#include <iostream.h>
#include <string.h>

//Write prototype for str_find function here

int main (){
  char string_1[80];      // Target string
  char string_2[5] = "This"; //The string to find
  int result;
  cin.getline(string_1,80);
  result = str_find(string_1, string_2);
  //str_find finds the occurrence of string_2 within
  //string_1
     if(result)
       cout<<"String Found";
     else
       cout<<"Not Found";
  return 0;
}

// Write function definition for str_find
```

Q8. A shop management system maintains information about different items in the shop. A single record contains information regarding item name, item number, item description, price and expiry date. Write the prototypes and definitions for the following functions [2+5]

a. **A function to input different records from the user.**
b. **A function that finds a record based on item's name and displays the complete information about the searched item.**

```
#include <iostream.h>
#include <string.h>

struct ITEM{
  char  name[20];
  int   number;
  char  description[50];
```

```
  float price;
  DATE expiry;
};
struct DATE{
  int month;
  int year;
}

// Write prototype for get_items function here
// Write prototype for display_item function here
int RECORDS = 10;
int main (){
  ITEM item[RECORDS];
  char item_to_find[20];

  cout<<"Please, enter an item name : ";
  cin>>item_to_find;
  get_items(item); // the function inputs 20 records
                   // from the user
  display_item(item_to_find);
  return 0;
}
// Write function definition for get_items function here
// Write function definition for display_item function here
```

Q9. Write the missing code snippets for the following program that inputs the current coordinates (x,y) of a user. It finds and displays the shortest distances from all possible destinations. [4]

```
#include <iostream.h>
#include <math.h>
void main(){
   //List of coordinates for all possible destinations
   int coordinates[7][2] ={{20,40},{100,60},{300,300},
   {400,50},{50,150},{360,360},{140,140}};
   int x_coord,y_coord;//current coordinate of the user
   cin>>x_coord>>y_coord;

   // Write your code here to find the shortest
   distances from (x_coord,y_coord) to all possible
   destinations.
}
```

```
Hint: useful functions from math.h
double sqrt(double);      double pow(double,double);
```