Homework 8 Questions

Q.1)

What will be the output of the following

```
#include<simplecpp>
struct Pixel
{
        int C, R;
};

void Display(Pixel P)
{
        cout << "Col "<< P.C << " Row " << P.R << endl;
}

main_program
{
        Pixel X = {40,50}, Y, Z;
        Z = X;
        X.C += 10;
        Y = Z;
        Y.C += 10;
        Y.R += 20;
        Z.C -= 15;
        Display(X);
        Display(Y);
        Display(Z);
}
```

Q.2) Correct the syntax errors

```
#include<simplecpp>
struct Pixels
{
        int color, style;
}
```

```
void showPoint(Pixels P)
{
        cout << P.color, P.style << endl;
}

main_program
{
        Pixels Point1 = (5, 3);
        showPoint(Point1);
        Pixels Point2 = Point1;
        color.Point1 += 2;
        showPoint(Point2);
}
```

Q.3) What will be the output?

```
#include<simplecpp>
int fun(int& a, int b)
{
        a += 3;
        b = 3;
        cout << "a=" << a << "b=" << b << "\n";
        return a%b;
}
main_program
{
        int a = 7;
        int b = 4;
        int c = fun (a,b);
        cout << "a=" << a << "b=" << b << "c=" << c << "\n";
}
```

Q.4) The following program takes an integer input from the user and generates a sequence of numbers related to that. Figure out what the sequence is.

```
#include<simplecpp>
void func(int a)
{
        int d[10];
        int new, size, i=0;
        while(a>0)
        {
                d[i]=a%10;
                i++;
        a/=10;
        }
        size=i;
        new=0;
        for(i=0;i<size;i++)
                new=new+d[i];
        cout<<new<<endl;
        if(new/10 > 0)
                func(new);
}
main_program
{
        int num;
        cout<<"Input an integer: "; cin>>num;
        func(num);
}
```

Q.5) Identify the error

```
#include<simplecpp>
void marksUpdate(student a)
{
        struct student
        {
                int roll;
                char name[30];
                int batch;
                float marks;
        };
        float change=0;

        cout << “What is the change in marks? “;
        cin>> change;
        a.marks+=change;
        cout<< “Updated marks are: “<<a.marks<<endl;
}

main_program
{
        student s;
        cout<<”Enter roll no.: “; cin>>s.roll;
        cout<<”Enter name: “; cin>>s.name;
        cout<<”Enter batch: “; cin>>s.batch;
        cout<<”Enter marks: “; cin>>s.marks;
        char option;
        cout<<”Is there a change in marks? (enter Y or N): “; cin>>option;
        if(option==’Y’ || option==’y’) marksUpdate(s);
}
```

Q.6) What will be the output of following code for the following sample input

```
Govind
Lahoti
Alan
John
Shraddha
Rana
Tanmoy
bhunia
Dhantu
buragohain
```

```
#include<simplecpp>
main_program{
          struct Name{                    //structure type definition
                 char Firstname[50];
                 char Lastname[50];
        };
        Name cs101_students[5];
        for(int i=0;i<5;i++)
        {
                 cin>>cs101_students[i].Firstname;
                 cin>>cs101_students[i].Lastname;
        }
        cout<<cs101_students[4].Firstname[4]<<endl;
        cout<<cs101_students[3].Firstname<<endl;
        cout<<cs101_students[2].Lastname[2]<<endl;
}
```

Q.7) The following program return a Disk having two given points as the endpoints of a diameter. Disk definition is same as taught in class(Chap17, slide number10). Fill in the blanks to complete the program.

```
#include<simplecpp>
struct Point{double x, y;};    // Point structure definition

Point midpoint(const Point &a,const Point &b){
        Point mp;
        mp.x = (a.x + b.x)/2;
        mp.y = (a.y + b.y)/2;
        return mp;
}

struct Disk{
        double radius;            // Disk structure definition
        Point center;
};
```

```
int main(){
        initCanvas();
        Point p={200,200},
        q={200,400};
        Disk d;
        d.radius=.......................................   // fill in the blank1
        d.center=midpoint(p,q);
        Circle c1(.......................................);  //fill in the blank2
        wait(10);
}
```

Q.8) The following program will find the smallest exact divisor of given number *n*. Obvious one is start a loop from 2,3,4…...n and check which will divide the n. But the following is more efficient in terms of checking the less number as compare to 2,3,......n. So what you have to do is your favorite fill the blanks.

```
#include<simplecpp>
int sdivisor(int n)
{
        int smallest,d,r;
         if(n%2==0)
                smallest=2;
        else{
                d=3;
                r=sqrt(n);
                while(..........................)      //blank1
                        d=d+2;
                if(n%d==0)
                        smallest=d;
                else smallest=.............;       //blank2
        }
        return smallest;
}
main_program{
        int n, res;
        cin>>n;
        res=sdivisor(n);
        cout<<res;
}
```

Q.9) Find the output of the following program. Assuming all the desired header files are already included, which are required to run the code.

```
#include<simplecpp>
struct Play
{
        int score, bonus;
};

void calculate(Play &P, int N)
{
        P.score++;
        P.bonus += N;
}

int main()
{
        Play PL = {10, 15};
        calculate(PL, 5);
        cout << PL.score << ":" << PL.bonus << endl;
        calculate(PL,10);
        cout << PL.score << ":" << PL.bonus << endl;
        calculate(PL, 15);
        cout << PL.score << ":" << PL.bonus << endl;
}
```

Q.10) Find the output of the following program. Assuming all the desired header files are already included, which are required to run the code.

```
#include<simplecpp>
struct MyBox
{
        int length, breadth, height;
};
void dimension (MyBox M)
{       cout << M.length << "x" << M.breadth << "x";
        cout << M.height << endl;
}

int main ()
{           MyBox B1 = {10, 15, 5}, B2, B3;
        ++B1.height;
        dimension(B1);
        B3 = B1;
        ++B3.length;
        B3.breadth++;
        dimension(B3);
        B2 = B3;
        B2.height += 5;
        B2.length--;
        dimension(B2);
}
```

Q.11) Fill in the blanks of the following code which stores the difference of two fractions as another fraction.

```
#include<simplecpp>
struct Fraction
{   int n;
  int d;
};
Fraction subtract(Fraction f1, Fraction f2)
{       int denom, num;
        Fraction frac;
        denom = f1.d * __a___;
        num = ((f2.n * -1) * __b__) - ((__c__ * -1) * f2.d);
        frac = {num, denom};
        return ___d___;
}
int main()
{
        Fraction f1,f2,Difference;
        cout << "Input the first numerator: ";
        cin >> f1.numerator;
        cout << "Input the first denominator: ";
        cin >> f1.denominator;
        // Allocate second fraction
```

```
    Fraction f2;
    cout << "Input the second numerator: ";
    cin >> f2.numerator;
    cout << "Input the second denominator: ";
    cin >> f2.denominator;
    Difference= subtract(f1, f2);
}
```

Q.12)

```
#include<simplecpp>

struct student {
    int rollNumber;
    char name[50];
    int year;
    char division;
};
struct marks {
    int rollNumber;
    int marksPhysics;
    int marksChemistry;
    int marksMaths;
};

main_program {
    marks m[10];
    student s[10] = {1,"Goyal",3,'A'};   //Statement 1
    int counter;

    for(counter = 0;counter<9;counter++) {
        s[counter+1] = OP ;    //Statement 2
    }

    for(counter = 0;counter<10;counter++) {
        m[counter].rollNumber = counter;
        m[counter].marksPhysics = counter+10;
        m[counter].marksChemistry = counter+20;
        m[counter].marksMaths = counter+30;
        }

        cout << m[9].rollNumber << ", " << m[9].marksPhysics << ", " <<
m[9].marksChemistry << ", " << m[9].marksMaths;  //Statement 3

}
```

Which of the following is/are correct?

(A) The statement 1 'student s[10] = {1,"Goyal",3,'A'};' assigns these values to all elements i.e. from s[0] to s[9]

(B) If OP is replaced with s[counter], then the program iteratively assigns values {1,"Goyal",3,'A'} from s[1] to s[9]

(C) The last cout statement, will print "9, 19, 29, 39"

(D) None of these

Q.13) You are given a list of nodes, the definition of a node is given below -

```
struct node
{
  char val;
  struct node *next;
};
```

Each node in the list is pointing to another node in the same list (by the pointer "struct node *next"). No node in the list can be pointed by more than one node.

For example, if you have only 3 nodes x,y,z, the list could be - x->y->z or y->x->z or z->y->x and so on… (x->y means x is pointing to y) But both x->y and z->y are not possible. The last node in the list has the value of "next" as NULL.

We need to write a function that will take two parameters

1. The start node of the list

2. An integer k

The function should return the ‘k-th’ element from the end of the list. Given the list a->b->c->d->e and k = 2 the function should return the 2nd last element 'd'

Given below are two different implementation of the function:(Assume input will be always valid i.e. list will not be NULL, k will always be less than the size of the list) -

```
//Implementation 1
char find_kth_last_elem(node *start, int k)
{
   node *ptr = start;
   int size = 0;
   int i=1;
   while(ptr != NULL)
   {
      ptr = ptr->next;
      size++;
   }
   ptr = start;
   while(i<=size-k)
   {
      ptr = ptr->next;
      i++;
   }
   return ptr->val;
}
```

```
//Implementation 2
char find_kth_last_elem(node *start, int k)
{
   node *ptr = start;
   node *kth_ptr = start;
   int size = 0;
   while(ptr != NULL)
```

```
    {
        if(size>=k)
            kth_ptr = kth_ptr->next;
        ptr = ptr->next;
        size++;
    }
    return kth_ptr->val;
}
```

Which of the implementation is/are correct?

Q.14)

In the following program fragment, assume that variables 'i', 'j', 'n', 'min', and 'temp' are declared to be of type 'int'. Similarly, assume that the array 'A' is declared as an integer array of size 10. Assume further that the number of elements in the array 'A' is given by 'n' (so that n <= 10).

```
for (i = 0; i < n-1 ; i++) {
  min = i;
  for (j = i + 1; j < n; j++){
    if (A[j] < A[min]){
      min = j;
    }
  }
  //swapping A[i] and A[min]
  temp = A[i];
  A[i] = A[min];
  A[min] = temp;
}
```

If the content of the array A before the for loop starts executing is {9,1,8,2,7,3,6,4,5,0}, and if the value of n is 10, what will be the contents of the array A after the fourth swap takes place? Note that A[0] = 9 and A[9] = 0 before the for loop starts executing?

Q.15)

Suppose "birthDays" is an unsorted array of integers, in which each element of the array is the date of birth of a student enrolled in CS101 in YYYYMMDD format. Assume that all birth dates in the array are in the year 1900 or later. Assume that months are represented by two decimal digits ranging from 01 through 12, and dates are represented by two decimal digits ranging from 01 through 31.

We want to sort the elements of "birthDays" in increasing order of months and dates, without worrying about the year. Thus, all birth dates in the month of January (MM is 01) should appear before all birth dates in the month of February (MM is 02) and so on in the sorted array. Among birth dates in the same month, we want a birth date with a smaller value of DD to appear before a birthday with a larger value of DD in the sorted array.

Thus, if 19710206, 20000201 and 19840107 are elements of the "birthDays" array, then 19840107 should appear before 20000201, which in turn should appear before 19710206 in the sorted array.

A student has written the following C++ function to sort the elements of "birthDays" in the order described above. Assume that the input integer parameter "n" is the size of the array, and all "n" birth dates are initially stored (perhaps in an unsorted order) in "birthDays" when the function "sortBirthDays" is called. You may also assume that n > 0.

```
void sortBirthDays(int birthDays[], int n)
{
        int someBirthDay;
        int index;

         do {
                index = n-1;
                for (int i = 0; i < n-1; i++) {
                        if (compare(birthDays[i], birthDays[index])) {
                        index = i;
                }
        }
        if (index < n-1) {
                someBirthDay = birthDays[index];
                birthDays[index] = birthDays[n-1];
                birthDays[n-1] = someBirthDay;
        }

        n--;
        } while (n > 1);
}
```

Which of the following “compare” functions will result in the desired sorting of “birthDays” array?

(A) bool compare(int date1, int date2) { return (date1 < date2); }

(B) bool compare(int date1, int date2) { return ((date1%10000) < (date2%10000)); }

(C) bool compare(int date1, int date2) { return ((date1%10000) >= (date2%10000)); }

(D) bool compare(int date1, int date2) { return ((date1%10000)/100 >= (date2%10000)/100); }

Q.16). Consider the following program.

```
#include <simplecpp>
struct scientist {
  char *name;
};
scientist sc;
scientist copy_struct() {
  sc.name = "Ramanujan";
  return sc;
}
main_program {
  scientist m = copy_struct();
  m.name = "Srinivsa";
  cout <<  m.name << endl;
  cout << sc.name << endl;
}
```

What will be the output of the following?

Select the appropriate choice(s)

A) Ramanujan
   Ramanujan

B) Srinivsa
   Ramanujan

C) Ramanujan
   Srinivsa
D) Srinivsa
   Srinivsa
E) None of these

Q.17) Consider the following program that calculates the penalty points based on the account type and the balance.

```
#include<simplecpp>

struct bankCustomer {
  long int account_number;
  char customer_first_name[50];
  double balance;
  char account_type;
};
main_program {
  bankCustomer customer = {123,"Amit",4850,'R'};
  int penalty_points = 10;
  if(customer.account_type  ==  'S') {
     if(customer.balance <=  0) {
        penalty_points = 10;
     }
     else {
        penalty_points = 0;
     }
  }
 if(customer.account_type  ==  'R') {
     if(customer.balance < 5000) {
        penalty_points = 40;
     }
     else {
        penalty_points = 0;
     }
  }
  if(penalty_points<0)
     cout << "Invalid account" << endl;
  if(penalty_points>0)
     cout << "Penalty is levied on the account" << endl;
  if(penalty_points == 0)
     cout << "Account valid. No penalty levied" << endl;
}
```

What would be the output produced by the program given above?

A) Invalid account

B) Penalty is levied on the account

C) Account valid. No penalty levied

D) None of these

Q.18) Consider the following code fragment and choose the correct option(s) from below.

```
#include <simplecpp>
void swap(int *a, int *b) {
   int *temp;
   temp = a;
   a = b;
   b = temp;
}
main_program {
   int A = 4, B = 6;
   swap(&A, &B);
   cout << A << B;

}
```

A) The program prints 46

B) The program prints 64

C) The swap function swaps the addresses of A and B

D) The swap function swaps the values of a and b

E) None of these

Q.19)

Describe what does the following programming computes without executing it.

```
#include<simplecpp>
void swap(char *x, char *y){
        char temp;
```

```
            temp = *x;
            *x = *y;
            *y = temp;
        }

        void fun(char *a, int l, int r){
            int i;
            if(l == r)
                cout << a <<endl;
            else{
                for(i = l; i <= r; i++){
                    swap((a+l), (a+i));
                    fun(a, l+1, r);
                    swap((a+l), (a+i));
                }
            }
        }

        int main(){

            char str[] = "CS101";
            int n = strlen(str);// strlen (library func) computes the length of the givesn
string
            fun(str, 0, n-1);
            return 0;
        }
```

Q.20)

Given a number x, the following program computes all the sequences (non increasing) such that sum of the series is equal to x.
example:

```
    x = 4
```

output:

```
    1 1 1 1
    2 1 1
    2 2
```

```
3 1
4
```

Fill in the blanks in the below program,

```
#include<simplecpp>
void print(int a[], int n){
    for(int i=0; i<n; i++)
        cout << a[i] <<" ";
    cout << endl;
}

void series_calc(int x, int arr[], int c_sum, int id){
    if(c_sum == x){
        print(arr, id);
        return;
    }
    int num = 1;
    while(num<=(x- c_sum) && (id==0 || num<= arr[id-1])){
        arr[id] = num;
        series_calc(____________);
        num++;
    }
}

int main(){
    int x = 4;
    int arr[x];
    series_calc(x, arr, 0, 0);
    return 0;
}
```