1. a) Write any two differences between array type and structure type. [2]

Array Type	Structure Type
• Collection of homogenous data type	• Collection of heterogenouis data type
• Array name is allocated memory	• Structure name is not allocated memory
• Passed by reference to a function	• passed by value/reference to a function
• Cannot use empty initializer	• Can use empty initializer

Select any two.

b) What is the role of an initializer in an array?
An initializer assigns values to the elements of an array, when the array is being created.
What happens when number values inside the initializer
i) is less than the number of elements in the array
Remaining elements will be 0 (zero) for int/float/double type and nul character for char type.
ii) is more than the number of elements in the array [2]
Compuiler will flag syntax error.

c) Write any two differences between buitl-in functions random() and randomize(). [2]

random()	randomize()
• Return type int	• Return type void
• Has an int type parameter	• Has no parameter

d) Write any three differences between macro identifier and constant identifier. [3]

Macro Identifier	Constant Identifier
• Created with compiler directive #define	• Created with keyword const
• Has no data type	• Has a data type
• No memory is allocated	• Memory is allocated
• Cannot use scope resolution operator with macro name	• Can use scope resolution operator with macro name
• Compiled code has no macro name	• Compiled code has constant name

Select any three.

e) Write a C++ expression to do the following: [3]
i) Generate a four digits positive random integer
```
1000+random(9000)
```
ii) Generate an uppercase random character
```
65+random(26)      //'A'+random(26)
```
iii) Generate a double digit negative integer
```
-10-random(90)     //-(10+random(10))
```

2. a) Rewrite the program after removing all the syntax errors (**underline the corrections**): [3]

```
#include<iostream.h>
void main()
{
   const MAX=10;
   int arr[MAX], s=0;
   //int MAX=10, arr[10], s=0;
   for (int x=0; x<MAX; x++)
   {
      int z=arr(x);
      s+=z;
   }
   cout<<s<<endl;
}
```

b) Write the output generated by the C++ program sement given below: [2]

```
char str[]="nOvEmbErDeCeMbER";
for (int k=0; str[k]; k++)
   if (str[k]>='A' && str[k]<='Z')
      str[k]++;
   else
      k%2 ? --str[k] : str[k]-=32;
cout<<str<<endl;
```

Output: `NPVFMaFqEdDdNaFS`

c) Write the output generated by the C++ program segment given below: [3]

```
int arr[8]={59, 77, 98, 39, 48, 99, 87, 65};
int s1=0, s2=0;
for (int x=0; x<8; x+=2)
{
   int a=arr[x], b=arr[x+1];
   s1+=a/10;
   s2+=b%10;
   arr[x]=a/10+b/10;
   arr[x+1]=a%10+b%10;
}
for (int k=0; k<8; k++)
   cout<<arr[k]<<'*';
cout<<s1<<'*'<<s2<<endl;
```

Output: `12*16*12*17*13*17*14*12*26*30`

3. Write C++ functions for following:

a) Sort an array of double using bubble sort. Array name and number of elements in the array are passed as parameters to the function. Return type of the function is **void**. [3]

```
void bubblesort(double a[], int n)
{
   for (int k=1; k<n; k++)
      for (int x=0; x<n-k; x++)
         if (a[x]>a[x+1])
         {
            double t=a[x]; a[x]=a[x+1]; a[x+1]=t;
         }
}
```

b) Insert an integer in an array of integers. Array name, number of elements currently in the array, integer to be inserted and the position for insertion are passed as parameters to the function. Assume array size is a macro identifier `SIZE`. Return type of the function is **void**. [3]

```
void arrayinsert(int a[], int &n, int pos, int item)
{
   if (n==SIZE)
      cout<<"Array Overflow\n";
   else
   {
      for (int k=n-1; k>=pos; k--)
         a[k+1]=a[k];
      a[pos]=item;
      n++;
```

```
   }
}
```

c) Count and display the number of even integers (integer divisible by 2), sum of even integers and average of even integers stored in an array of integers. Array name and number of elements in the array are passed as parameters to the function. Return type of the function is **void**. [3]

```
void countsumavg(int a[], int n)
{
   int s=0, c=0;              //double s=0; int c=0;
   for (int k=0; k<n; k++)
      if (a[k]%2==0)
      {
         s+=a[k]; c++;
      }
   double avg=double(s)/n;   //double avg=s/n;
   cout<<"Count="<<c<<"\nSum="<<s<<"\nAverage="<<avg<<endl;
}
```

d) To check whether a string is a Palindrome or not. The string that is to be checked is passed as a parameter to the function. Return type of the function is **void**. [3]

Input: NITIN
Output: Palindrome
Input: NUTAN
Output: Not Palindrome

```
void chkstrpalin(char str[])
{
   int len=0;
   while (str[len])
      len++;
   int ri=len-1, palin=1;
   for (int le=0; le<ri && palin==1; le++, ri--)
      if (str[le]!=str[ri])
         palin=0;
   /*
      int palin=1;
      for (int k=0; k<len/2 && palin==1; k++)
         if (str[k]!=str[len-k-1])
            palin=0;
      OR
      int le=0, ri=len-1, palin=1;
      while (le<ri && palin==1)
         if (str[le++]!=str[ri--])
            palin=0;
      OR
      int k=0, palin=1;
      while (k<len/2 && palin==1)
      {
         if (str[k]!=str[len-k-1])
            palin=0;
         k++;
      }
   */
   if (palin==1)
      cout<<str<<" Palindrome\n";
```

```
      else
         cout<<str<<" Not Palindrome\n";
   }
```

4. Create a structure `employee` with following data members: [4]

   `eno` Employee Number (int type)
   `name` Employee Name (string type)
   `basic` Basic Salary (double type)
   `hrent` House Rent (double type)
   `perks` Extra allowance (double type)
   `gross` Gross Salary (double type)

   - Write a function `empinput()`, to input `eno`, `name` and `basic` of an `employee` passed as a parameter to the function.
   - Write a function `empsalary()`, to calculate `hrent`, `gsal` and `perks` of an `employee` passed as a parameter to the function. Data members `hrent` and `perks` are calculated according to the table given below:

Basic Salary	**House Rent**	**Extra allowance**
>=100000	30% of Basic Salary	15% of Basic Salary
>100000 and <=400000	25% of Basic Salary	10% of Basic Salary
>400000	20% of Basic Salary	5% of Basic Salary

   Gross Salary = Basic Salary + House Rent + Extra allowance
   - Write a `main()` function to create a variable of the type `employee`, call the function `empinput()`, `empsalary()` and display values stored in the variable of the type `employee`.

```
#include<iostream.h>
#include<stdio.h>
struct employee
{
   int eno; char name[20]; double basic, hrent, perks, gross;
};
void empinput(employee &a)
{
   cout<<"Code? "; cin>>a.eno;
   cout<<"Name? "; gets(a.name);
   cout<<"Basic? "; cin>>a.basic;
}
void empsalary(employee &a)
{
   if (a.basic<=100000)
   {
      a.hrent=0.30*a.basic;
      a.perks=0.15*a.basic;
   }
   else
   if (a.basic<=400000)
   {
      a.hrent=0.25*a.basic;
      a.perks=0.10*a.basic;
   }
   else
   if (a.basic<=400000)
   {
      a.hrent=0.20*a.basic;
      a.perks=0.05*a.basic;
```

```
    }
    a.gross=a.basic+a.hrent+a.perks;
}
void main()
{
    employee emp;
    empinput(emp);
    empsalary(emp);
    cout<<"Code ="<<emp.eno<<endl;
    cout<<"Name ="<<emp.name<<endl;
    cout<<"Basic="<<emp.basic<<endl;
    cout<<"HRent="<<emp.hrent<<endl;
    cout<<"Perks="<<emp.perks<<endl;
    cout<<"Gross="<<emp.gross<<endl;
}
```

5. Create a structure `mobile` with following members:

`model`	model name (string of 20 characters)
`storage`	internal storage (string type like "16 GB", "32 GB", "64 GB", …)
`mpix`	mega pixel (double type)
`price`	price (double type)

Write a C++ function to display details of all mobiles where internal storage is 32 GB and mega pixel exceeds 16 from an array of `mobile`. Array name, number of elements are passed as parameters to the function. At the end display number of such mobiles found. Return type of the function is **void**. [4]

```
struct mobile
{
   char model[20], storage[8]; double mpix, price;
};
void count(mobile a[], int n)
{
   int c=0;
   for (int k=0; k<n; k++)
      if (strcmpi(a[k].storage, "32 GB")==0 && a[k].mpix>16)
      {
         cout<<a[k].model<<","<<a[k].storage<<","
             <<a[k].mpix<<","<<a[k].price<<endl;
         c++;
      }
   cout<<"Storage=32 GB and MPix>16="<<c<<endl;
}
```