Assignment - II

Q1.
```
#include<stdio.h>
int main()
{ int arr[10];
  int i, size, min, max;
  printf("Enter size of array");
  scanf("%d", &size);
  printf("Enter elements in the array: ");
  ~~Scanf("%d"~~, for(i=0; i<size; i++)
    {scanf(" %d", &arr[i]); }
  max = arr[0];
  min = arr[0];
  for(i=1; i<size; i++)
  {if (arr[i] > max)
   {max = arr[i]; }
   if(arr[i]<min)
   {min = arr[i]; }
 }
printf("Maximum element: %d", max);
printf("Minimum element: %d", min);
}return0;
}
```

Q2.

	COMPILER	INTERPRETER
INPUT	It takes an entire program at a time	It takes a single line of code or instruction at a time.
OUTPUT	It generates intermediate object code	It does not produce any intermediate object code.
SPEED	Comparatively Faster	Slower.
ERROR Detection	Difficult	Easier Comparatively.

Q3.
a. In C programming, an array of pointer is an indexed set of variables, where the variable are pointers.
Pointers are an important tool in CS for creating, using and destroying all types of data structures. An array of pointers is useful for the same ~~se~~ reason that all arrays are useful : it allows you to numerically index a set of variables.

```
#include<stdio.h>
const int Array_Size[=5;
main()
{int array_of_integers[]={5,10,20,40,80};
 int i,*array_of_pointers[Array_Size];
 for(i=0; i<~~Siz~~Array_Size; i++)
 {array_of_pointers[i]=&array_of_integer[i];}
 for(i=0;i<Array_Size;i++)
 {printf("array_of_integers [%d]=%d\n",i,*array_of_pointers[i]);
 }
}
```

b Pointer to an array is also known as array ~~member~~ pointer.
We are using the pointer to access the components of the array.
We use parenthesis to pronounce pointer to an array.

```
#include<stdio.h>
main()
{ int(*a)[5];
   int b[5]={1,2,3,4,5};
   int i=0;
   a=&b;
   for(i=0; i<5; i++)
   {printf("%d",*(*a+i));}
}
```

Q4.
```
#include<stdio.h>
main()
{ int arr1[50], arr2[50], arr3[50], m, n, i, J, k=0;
  printf("Enter Size of array1:");
  scanf("%d", &m);
  printf("Enter array1 elements:");
  for(i=0; i<m; i++)
  { scanf("%d", &arr1[i]); }
  printf("Enter Size of array2:");
  Scanf(" %d", &n);
  printf("Enter array2 elements:");
  ~~Scanf("~~for(J=0; J<n; J++)
   { scanf("%d", &arr2[J]); }
  i=0; J=0;
  while (i<m &&J<n)
  {if (arr1[i] <arr2[J])
   { arr3[k]= arr1[i];
     i++;
   }
   else
   {arr3[k] = arr2[J];
    J++; }
   k++;
  }
  if(i>=m)
  {while (J<n)
   {arr3[k] =arr2[J];
    J++; k++; }
  }
printf("After Merging:");
for(i=0; i<m+n ; i++)
{printf("%d", arr3[i]); }
}
```

Q5. An array of structures in C can be defined as the collection of multiple structure variables contains information about different entities. The array of structure in C are used to store information about multiple entities of different data types.

```
#include<stdio.h>
#include<string.h>
struct student
{ int rollno;
  char name[50];
};
main()
{int i;
 struct student st[5];
 printf("Enter Record of 5 Students:");
 for(i=0; i<5; i++)
 {printf("Rollno:");
  scanf("%d",&st[i].rollno);
  printf("Name:");
  scanf("%s", st[i].name);
 }
~~fo~~for(i=0, i<5; i++)
{printf("Rollno: %d, Name: %s",st[i].rollno,st[i].name);}
}
```