```c
//Set Operations on Arrays

struct Array
{
 int A[10];
 int size;
 int length;
};

void Display(struct Array arr)
{
 int i;
 printf("\nElements are\n");
 for(i=0;i<arr.length;i++)
 printf("%d ",arr.A[i]);
}

struct Array* Union(struct Array *arr1,struct Array *arr2)
{
 int i,j,k;
 i=j=k=0;

struct Array *arr3=(struct Array *)malloc(sizeof(struct Array));

 while(i<arr1->length && j<arr2->length)
 {
 if(arr1->A[i]<arr2->A[j])
 arr3->A[k++]=arr1->A[i++];
 else if(arr2->A[j]<arr1->A[i])
 arr3->A[k++]=arr2->A[j++];
 else
 {
 arr3->A[k++]=arr1->A[i++];
 j++;
 }
 }

 for(;i<arr1->length;i++)
 arr3->A[k++]=arr1->A[i];
```

```
 for(;j<arr2->length;j++)
 arr3->A[k++]=arr2->A[j];

 arr3->length=k;
 arr3->size=10;

 return arr3;
}

struct Array* Intersection(struct Array *arr1,struct Array* arr2)
{
 int i,j,k;
 i=j=k=0;

 struct Array *arr3=(struct Array *)malloc(sizeof(struct Array));

 while(i<arr1->length && j<arr2->length)
 {
 if(arr1->A[i]<arr2->A[j])
 i++;
 else if(arr2->A[j]<arr1->A[i])
 j++;
 else if(arr1->A[i]==arr2->A[j])
 {
 arr3->A[k++]=arr1->A[i++];
 j++;
 }
 }

 arr3->length=k;
 arr3->size=10;

 return arr3;
}

struct Array* Difference(struct Array *arr1,struct Array* arr2)
{
 int i,j,k;
 i=j=k=0;
```

```
 79:
 80:  struct Array *arr3=(struct Array *)malloc(sizeof(struct Array));
 81:
 82:  while(i<arr1->length && j<arr2->length)
 83:  {
 84:  if(arr1->A[i]<arr2->A[j])
 85:  arr3->A[k++]=arr1->A[i++];
 86:  else if(arr2->A[j]<arr1->A[i])
 87:  j++;
 88:  else
 89:  {
 90:  i++;
 91:  j++;
 92:  }
 93:  }
 94:  for(;i<arr1->length;i++)
 95:  arr3->A[k++]=arr1->A[i];
 96:
 97:
 98:  arr3->length=k;
 99:  arr3->size=10;
100:
101:  return arr3;
102: }
103:
104: int main()
105: {
106:  struct Array arr1={{2,9,21,28,35},10,5};
107:  struct Array arr2={{2,3,9,18,28},10,5};
108:  struct Array *arr3;
109:  arr3=Union(&arr1,&arr2);
110:  Display(*arr3);
111:  arr3=Intersection(&arr1,&arr2);
112:  Display(*arr3);
113:  arr3=Difference(&arr1,&arr2);
114:  Display(*arr3);
115:
116: return 0;
117: }
```