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;

    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])
            j++;
        else
        {
            i++;
```

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


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

    return arr3;
}

int main()
{
    struct Array arr1={{2,9,21,28,35},10,5};
    struct Array arr1={{2,3,9,18,28},10,5};
    struct Array *arr3;

arr3=Union(&arr1,&arr2);
Display(*arr3);

return 0;
}
```