Assignment - 15 A Job Ready Bootcamp in C++, DSA and IOT MySirG

Array and Functions in C Language

1. Write a function to find the greatest number from the given array of any size. (TSRS)

Code

#include<stdio.h>

int greatest\_num\_in\_array (int [] , int);

int main() {

    int i,j, a[1000],n;

    printf("Enter size of array :\n");

    scanf("%d",&n);

    printf("Enter %d numbers :\n",n);

    for(i=0;i<n;i++)

    scanf("%d",&a[i]);

  printf("greatest number : %d",greatest\_num\_in\_array(a,n));

    return 0;

}

int greatest\_num\_in\_array (int b[],int n)

  {

    int  i,j ;

    for(i=0;i<n;i++)

    {

       if(b[0]<b[i])

       {

        b[0]=b[i];

       }

    }

    return b[0];

  }

Output

Enter size of array :

4

Enter 4 numbers :

4 3 5 2

greatest number : 5

1. Write a function to find the smallest number from the given array of any size. (TSRS)

Code

#include<stdio.h>

int smallest\_num\_in\_array (int [] , int);

int main() {

    int i,j, a[1000],n;

    printf("Enter size of array :\n");

    scanf("%d",&n);

    printf("Enter %d numbers :\n",n);

    for(i=0;i<n;i++)

    scanf("%d",&a[i]);

  printf("smallest number : %d",smallest\_num\_in\_array(a,n));

    return 0;

}

int smallest\_num\_in\_array (int b[],int n)

  {

    int  i,j ;

    for(i=0;i<n;i++)

    {

       if(b[0]>b[i])

       {

        b[0]=b[i];

       }

    }

    return b[0];

  }

Output

Enter size of array :

4

Enter 4 numbers :

4 1 3 2

smallest number : 1

1. Write a function to sort an array of any size. (TSRS)

Code

#include<stdio.h>

int sort\_array( int [] , int );

int main() {

    int  i, a[1000], n;

    printf("Enter size of array :\n");

    scanf("%d",&n);

    printf("Enter %d numbers :\n",n);

    for(i=0;i<n;i++)

    scanf("%d",&a[i]);

    sort\_array(a,n);

    return 0;

}

int sort\_array( int b[] , int n)

{

     int  i,j;

    for(i=0;i<n;i++)

    {

         for(j=0;j<n;j++)

         {

              if(b[i]<b[j])

              {

                b[i] = b[i] + b[j];

                b[j] = b[i] - b[j];

                b[i] = b[i] - b[j];

              }

         }

    }

  for(i=0;i<n;i++)

  printf(" %d ",b[i]);

}

Output

Enter size of array :

4

Enter 4 numbers :

3 4 2 1

1 2 3 4

4. Write a function to rotate an array by n position in d direction. The d is an indicative

value for left or right. (For example, if array of size 5 is [32, 29, 40, 12, 70]; n is 2 and

d is left, then the resulting array after left rotation 2 times is [40, 12, 70, 32, 29] )

Code

#include<stdio.h>

int rotate\_array( int , int, int , int[] );

int main() {

    int  i, a[1000], n,d,s;

    printf("Enter size of array :\n");

    scanf("%d",&s);

    printf("Enter position of array :\n");

    scanf("%d",&n);

    printf("Enter direction of array for left 1 or for rignt 2  :\n");

      scanf("%d",&d);

   printf("Enter array :\n");

    for(i=0;i<s;i++)

    scanf("%d",&a[i]);

    rotate\_array(n,d,s,a);

    return 0;

}

int rotate\_array ( int n , int d , int s ,int b[])

{

     int  i,j,temp[100];

   if(d == 1)

   {

  for(i=0;i<n;i++)

  temp[i] = b[i];

    for(i=0;i<n;i++)

    {

      for(j=0;j<s;j++)

      {

         b[j]=b[j+1];

      }

    }

    for(i=0;i<n;i++)

    {

       b[s-n+i] = temp[i];

    }

  for(i=0;i<s;i++)

  printf(" %d ",b[i]);

   }

   if(d == 2)

   {

  for(i=0;i<n;i++)

  temp[i] = b[s-n+i];

    for(i=0;i<n;i++)

    {

      for(j=0;j<s;j++)

      {

         b[s-n-j+1]=b[s-n-j];

      }

    }

    for(i=0;i<n;i++)

    {

       b[i] = temp[i];

    }

  for(i=0;i<s;i++)

  printf(" %d ",b[i]);

   }

}

Output

Enter size of array :

5

Enter position of array :

2

Enter direction of array for left 1 or for rignt 2 :

2

Enter array :

1 2 3 4 5

4 5 1 2 3

5. Write a function to find the first occurrence of adjacent duplicate values in the array.

Function has to return the value of the element.

Code

#include<stdio.h>

int first\_dublicate\_adjacent\_value\_in\_array (int [] , int);

int main() {

    int i,j, a[1000],n;

    printf("Enter size of array :\n");

    scanf("%d",&n);

    printf("Enter %d numbers :\n",n);

    for(i=0;i<n;i++)

    scanf("%d",&a[i]);

 printf("the first occurrence of adjacent duplicate values in the array : %d",(first\_dublicate\_adjacent\_value\_in\_array(a,n)));

    return 0;

}

int first\_dublicate\_adjacent\_value\_in\_array (int b[],int n)

  {

    int  i,j,count=0 ;

    for(i=0;i<n-1;i++)

  {

      if(b[i]==b[i+1])

      {

         return b[i];

         break;

      }

  }

  }

Output

Enter size of array :

6

Enter 6 numbers :

1 2 2 3 3 4

the first occurrence of adjacent duplicate values in the array : 2

6. Write a function in C to read n number of values in an array and display it in reverse

order.

Code

#include<stdio.h>

void reverse\_order(int [], int);

int main() {

    int i, a[10],n=0;

     printf("Enter size of array :\n");

    scanf("%d",&n);

    printf("Enter %d numbers :\n",n);

    for(i=0;i<n;i++)

    scanf("%d",&a[i]);

     reverse\_order(a,  n);

    return 0;

}

void reverse\_order(int b[], int n)

  {

    int i;

    for(i=n-1;i>=0;i--)

    printf("%d ",b[i]);

   }

Output

nter size of array :

4

Enter 4 numbers :

1 2 3 4

4 3 2 1

7. Write a function in C to count a total number of duplicate elements in an array.

Code

#include<stdio.h>

int count\_dublicate\_num\_in\_array (int [] , int);

int main() {

    int i,j, a[1000],n;

    printf("Enter size of array :\n");

    scanf("%d",&n);

    printf("Enter %d numbers :\n",n);

    for(i=0;i<n;i++)

    scanf("%d",&a[i]);

  printf("Dublicate numbers : %d",(count\_dublicate\_num\_in\_array(a,n)-n)/2);

    return 0;

}

int count\_dublicate\_num\_in\_array (int b[],int n)

  {

    int  i,j,count=0 ;

    for(i=0;i<n;i++)

 for(j=0;j<n;j++)

    {

       if(b[i]==b[j])

       {

        count++;

       }

    }

    return count;

  }

Output

Enter size of array :

8

Enter 8 numbers :

1 2 3 4 1 2 3 6

Dublicate numbers : 3

8. Write a function in C to print all unique elements in an array.

Code

#include<stdio.h>

void count\_dublicate\_num\_in\_array (int [] , int);

int main() {

    int i,j, a[1000],n;

    printf("Enter size of array :\n");

    scanf("%d",&n);

    printf("Enter %d numbers :\n",n);

    for(i=0;i<n;i++)

    scanf("%d",&a[i]);

      count\_dublicate\_num\_in\_array(a,n);

//  printf("Dublicate numbers : %d",(count\_dublicate\_num\_in\_array(a,n)-n)/2);

    return 0;

}

void count\_dublicate\_num\_in\_array (int b[],int n)

  {

    int  i,j,count=0 ;

    for(i=0;i<n;i++)

  {

      for(j=0;j<n;j++)

    {

       if(b[i]==b[j])

       {

        count++;

       }

    }

     if(count==1)

        {

            printf(" %d ",b[i]);

        }

    count=0;

  }

  }

Output

Enter size of array :

7

Enter 7 numbers :

1 1 2 3 4 4 5

2 3 5

9. Write a function in C to merge two arrays of the same size sorted in descending

order.

Code

#include<stdio.h>

int merge\_two\_array\_with\_desending\_sorting( int [] , int , int [] , int);

int main() {

    int  i, a1[1000],a2[1000], n1,n2;

    printf("Enter size of array[1] :\n");

    scanf("%d",&n1);

    printf("Enter %d numbers of array[1] :\n",n1);

    for(i=0;i<n1;i++)

    scanf("%d",&a1[i]);

    printf("Enter size of array[2] :\n");

    scanf("%d",&n2);

    printf("Enter %d numbers of array[2] :\n",n2);

    for(i=0;i<n2;i++)

    scanf("%d",&a2[i]);

    merge\_two\_array\_with\_desending\_sorting(a1,n1 ,a2,n2);

    return 0;

}

int merge\_two\_array\_with\_desending\_sorting( int b1[] , int n1, int b2[] , int n2)

{

    int s[10000],i ;

    for(i=n1;i<n1+n2;i++)

      {

           b1[i] = b2[i-n1];

      }

         printf("the merged Array\n");

         for(i=0;i<n1+n2;i++)

         printf(" %d ",b1[i]);

         printf("\n\n");

     int  j;  // int i again redeclaration error

    for(i=0;i<n1+n2;i++)

    {

         for(j=0;j<n1+n2;j++)

         {

              if(b1[i]>b1[j])

              {

                b1[i] = b1[i] + b1[j];

                b1[j] = b1[i] - b1[j];

                b1[i] = b1[i] - b1[j];

              }

         }

    }

  printf("the sorted Array\n");

  for(i=0;i<n1+n2;i++)

  printf(" %d ",b1[i]);

}

Output

Enter size of array[1] :

3

Enter 3 numbers of array[1] :

2 5 6

Enter size of array[2] :

4

Enter 4 numbers of array[2] :

2 6 4 7

the merged Array

2 5 6 2 6 4 7

the sorted Array

7 6 6 5 4 2 2

10. Write a function in C to count the frequency of each element of an array.

Code

#include<stdio.h>

void count\_dublicate\_num\_in\_array (int [] , int);

int main() {

    int i,j, a[1000],n;

    printf("Enter size of array :\n");

    scanf("%d",&n);

    printf("Enter %d numbers :\n",n);

    for(i=0;i<n;i++)

    scanf("%d",&a[i]);

      count\_dublicate\_num\_in\_array(a,n);

//  printf("Dublicate numbers : %d",(count\_dublicate\_num\_in\_array(a,n)-n)/2);

    return 0;

}

void count\_dublicate\_num\_in\_array (int b[],int n)

  {

    int  i,j,count=0 ;

    for(i=0;i<n;i++)

  {

      for(j=0;j<n;j++)

    {

       if(b[i]==b[j])

       {

        count++;

       }

    }

            printf("%d has a frequency : %d \n",b[i], count);

    count=0;

  }

  }

Output

Enter size of array :

5

Enter 5 numbers :

1 1 2 2 3

1 has a frequency : 2

1 has a frequency : 2

2 has a frequency : 2

2 has a frequency : 2

3 has a frequency : 1