# Data structure lab

# Operations on 1D array

Megha Patel

20124107

#include<stdio.h>

#define max 30

int traverse(int arr[],int num);

int insertatend(int arr1[],int num1);

int insertatbegin(int arr2[],int num2);

int insertatgiven(int arr3[],int num3);

int insertinsort(int arr4[],int num4);

int deletefromend(int arr5[],int num5);

int deletefrombegin(int arr6[],int num6);

int deletegiven(int arr7[],int num7);

int concatenation(int arr8[],int num8);

int invert(int arr11[],int num11);

int main()

{

int c,ch,ch1,a[max],n,i;

char ch3[2];

printf("enter size of array u want enter\n");

scanf("%d",&n);

printf("now enter them \n");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

do

{

printf("select the operations for the array\n");

printf("\n 1.traverse\n2.insertion\n3.delete\n4.concatenate\n5.inversion\n");

scanf("%d",&c);

switch(c)

{

case 1:

traverse(a,n);

break;

case 2:

printf("\n1.insert at end of the array\n2.insert at begiining\n3insert at given location\n4.insert in sorted array\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

/\* printf("enter size of array\n");

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}\*/

insertatend(a,n);

break;

case 2:

/\* printf("enter size of array\n");

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}\*/

insertatbegin(a,n);

break;

case 3:

insertatgiven(a,n);

break;

case 4:

/\* printf("enter size of array\n");

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}\*/

insertinsort(a,n);

break;

default:

printf("invalid");

break;

}

break;

case 3:

printf("\n1.delete from end of the array\n2.delete from begiining\n3delete at given location\n");

scanf("%d",&ch1);

switch(ch1)

{

case 1:

deletefromend(a,n);

break;

case 2:

deletefrombegin(a,n);

break;

case 3:

deletegiven(a,n);

break;

default:

printf("invalid");

}

break;

case 4:

concatenation(a,n);

break;

case 5:

invert(a,n);

break;

default:

printf("invalid choice\n");

break;

}

printf("\nWant to check more ' y ' for yes 'n' for no\n");

scanf("%s",ch3);}

while(ch3=='y');

}

int traverse(int a[],int num)

{

int i;

//int size=sizeof(a);

printf("traversed array is\n");

for(i=0;i<num;i++)

{

printf("%d",a[i]);

}

}

int insertatend(int arr1[],int num1)

{

int i,d;

if(num1==max)

{

printf("overflow!!!\n");

return 0;

}

//num1=num1+1;

printf("enter data to be inserted\n");

scanf("%d",&arr1[num1]);

num1=num1+1;

arr1[num1]=d;

//traverse(arr1,num1);

for(i=0;i<=num1-1;i++)

{

printf("%d",arr1[i]);

}

}

int insertatbegin(int arr2[],int num2)

{

int i,d,k;

//num1=num1+1;

if(num2==max)

{

printf("overflow!!!\n");

return 0;

}

printf("enter data to be inserted\n");

scanf("%d",&d);

for(k=num2;k>=0;k--)

{

arr2[k+1]=arr2[k];

}

arr2[0]=d;

printf("new array after insertion is\n");

for(i=0;i<=num2;i++)

{

printf("%d\n",arr2[i]);

}

}

int insertatgiven(int arr3[],int num3)

{

int i,d,k,loc;

//num1=num1+1;

if(num3==max)

{

printf("overflow!!!");

return 0;

}

printf("enter data to be inserted\n");

scanf("%d",&d);

printf("enter location to insert data\n");

scanf("%d",&loc);

for(k=num3;k>=loc;k--)

{

arr3[k+1]=arr3[k];

}

arr3[loc]=d;

printf("new array after insertion is\n");

for(i=0;i<=num3;i++)

{

printf("%d\n",arr3[i]);

}

}

int insertinsort(int arr4[],int num4)

{

int i,d,k,loc;

//num1=num1+1;

if(num4==max)

{

printf("overflow!!!");

return 0;

}

printf("enter data to be inserted\n");

scanf("%d",&d);

k=0;

while(arr4[k]<d)

{

loc=k+1;

k+=1;

}

for(k=num4;k>=loc;k--)

{

arr4[k+1]=arr4[k];

}

arr4[loc]=d;

printf("array after insertion\n");

for(i=0;i<=num4;i++)

{

printf("%d\n",arr4[i]);

}

}

int deletefromend(int arr5[],int num5)

{

int i;

if(num5==0)

{

printf("underflow!!");

return 0;

}

num5=num5-1;

arr5[num5]=NULL;

printf("after deltion\n");

for(i=0;i<num5;i++)

{

printf("%d\n",arr5[i]);

}

}

int deletefrombegin(int arr6[],int num6)

{

int k,i;

if(num6==0)

{

printf("underflow!!");

return 0;

}

for(k=0;k<=num6;k++)

{

arr6[k]=arr6[k+1];

}

arr6[num6]=NULL;

num6=num6-1;

printf("after deltion\n");

for(i=0;i<num6;i++)

{

printf("%d\n",arr6[i]);

}

}

int deletegiven(int arr7[],int num7)

{

int i,d,k,loc;

//num1=num1+1;

if(num7==0)

{

printf("underflow!!");

return 0;

}

//printf("enter data to be deleted");

//scanf("%d",&d);

printf("enter data location from data to be deleted");

scanf("%d",&loc);

k=0;

while(k<num7)

{

if(k==loc)

{

arr7[k]=arr7[k+1];

}

k+=1;

}

num7=num7-1;

arr7[num7]=NULL;

printf("after deltion\n");

for(i=0;i<num7;i++)

{

printf("%d\n",arr7[i]);

}

}

int concatenation(int arr8[],int num8)

{

int arr9[100],arr10[100],num9;

int i,j,k,l,size;

/\*printf("no. elements of array1\n");

scanf("%d",&num8);

for(i=0;i<num8;i++)

{

scanf("%d",&arr8[i]);

}\*/

printf("no. elements of array2\n");

scanf("%d",&num9);

printf("enter elemnt you want to conc with array 1\n");

for(j=0;j<num9;j++)

{

scanf("%d",&arr9[j]);

}

size=num9+num8;

for(k=0;k<=num8;k++)

{

arr10[k]=arr8[k];

}

k=0;

for(l=num8;l<size;l++,k++)

{

arr10[l]=arr9[k];

}

printf("after conc of 2 array\n");

for(i=0;i<size;i++)

{

printf("%d",arr10[i]);

}

}

int invert(int arr11[],int num11)

{

int i,t,j,k;

i=0;

j=num11;

while(i<j)

{

t=arr11[i];

arr11[i]=arr11[j];

arr11[j]=t;

i+=1;

j-=1;

}printf("inverted array is\n");

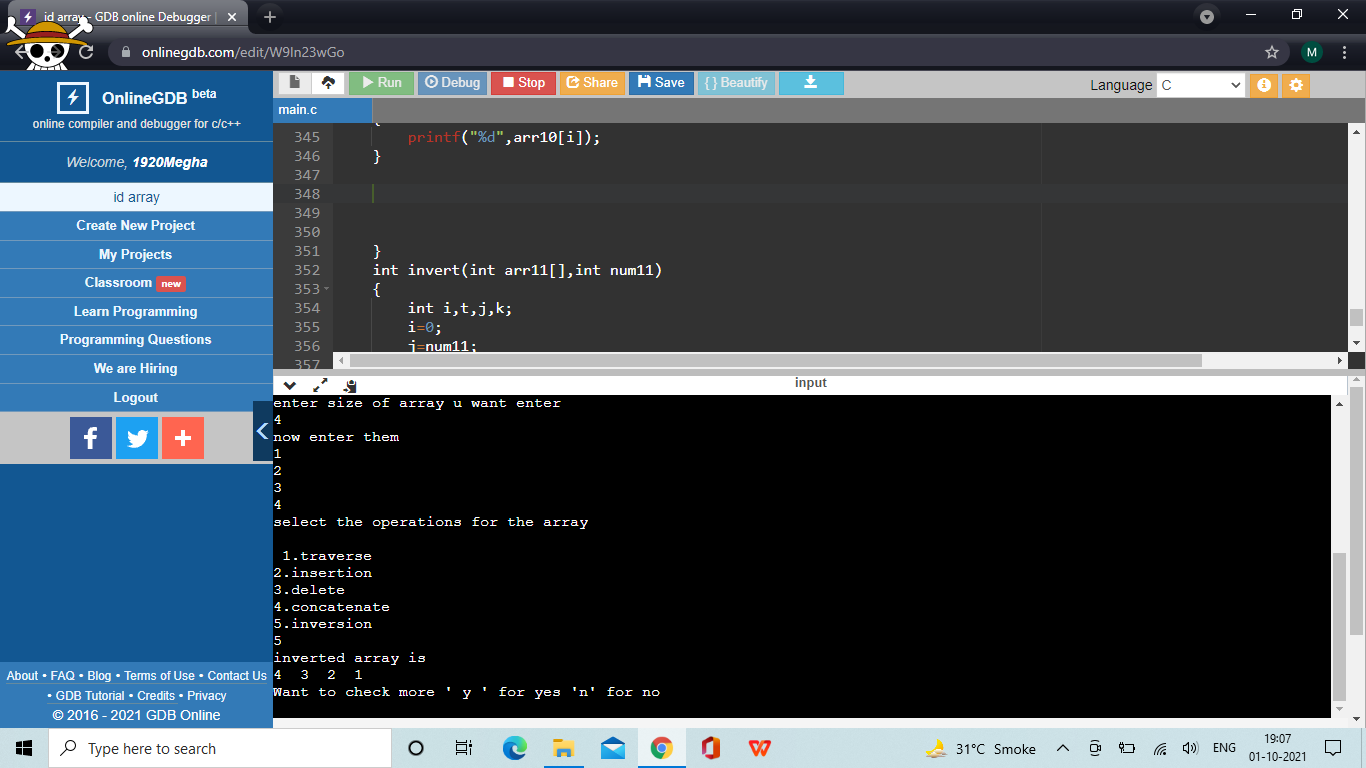
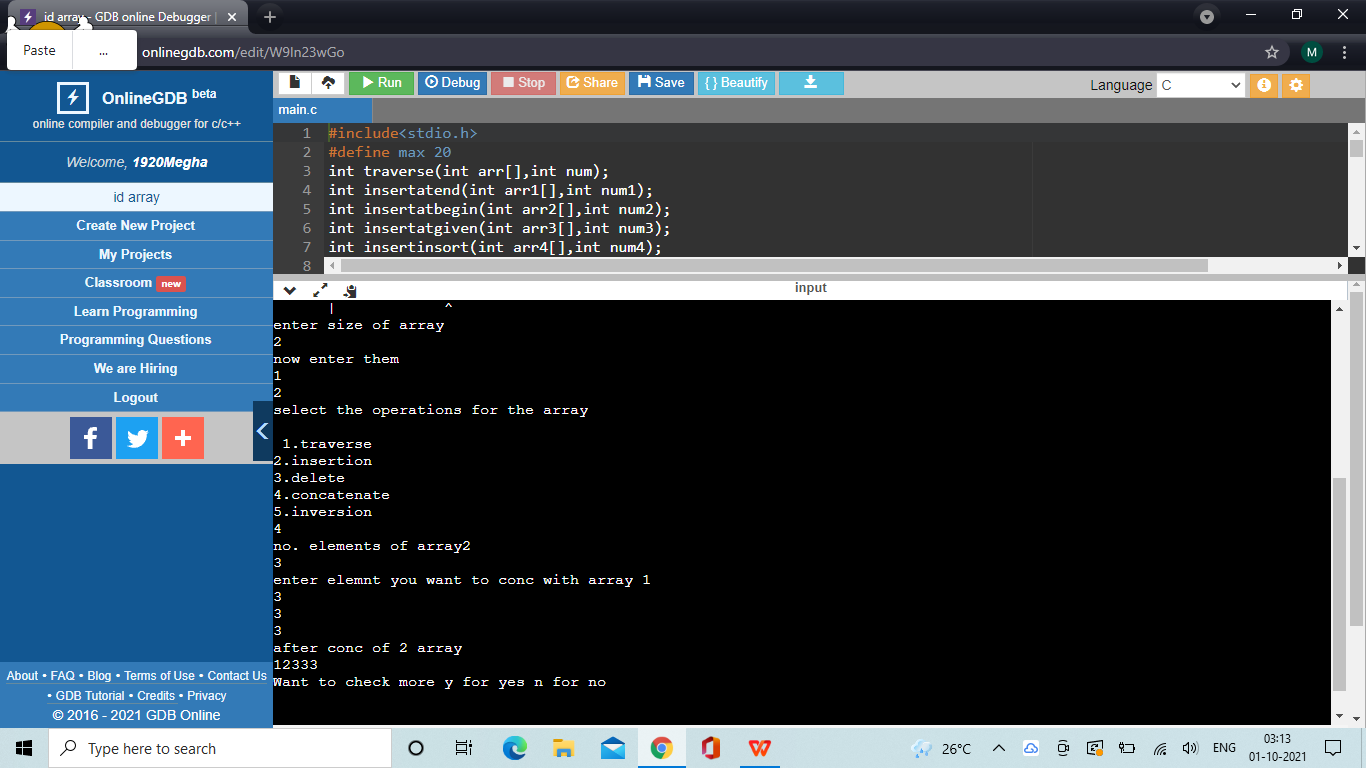
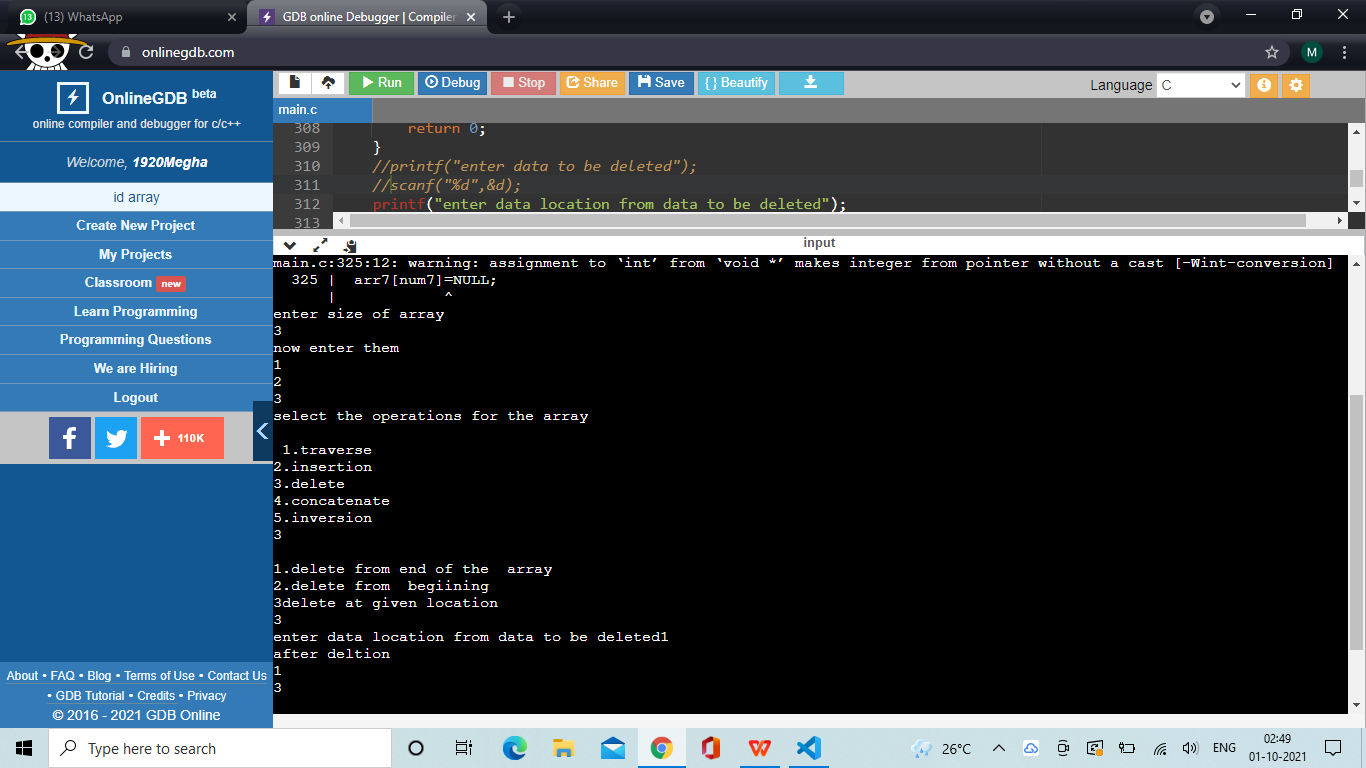
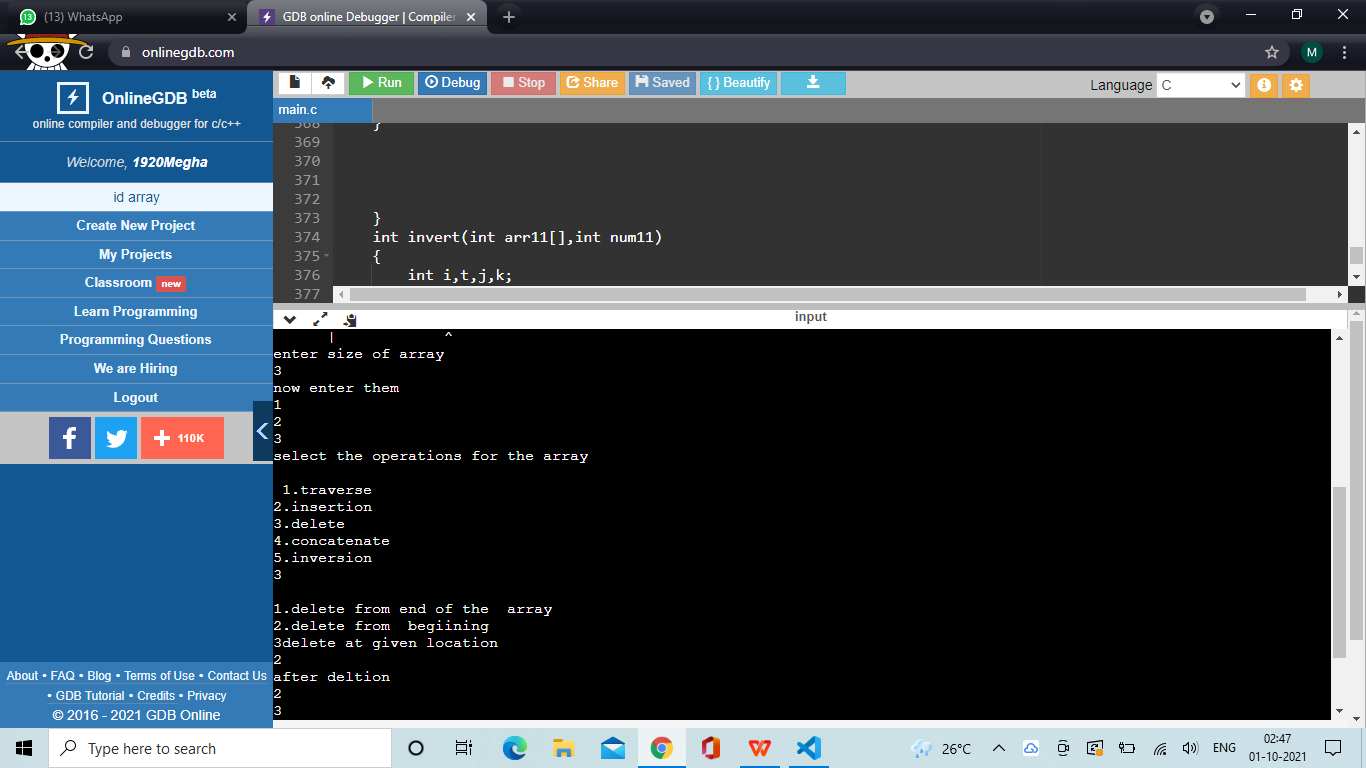
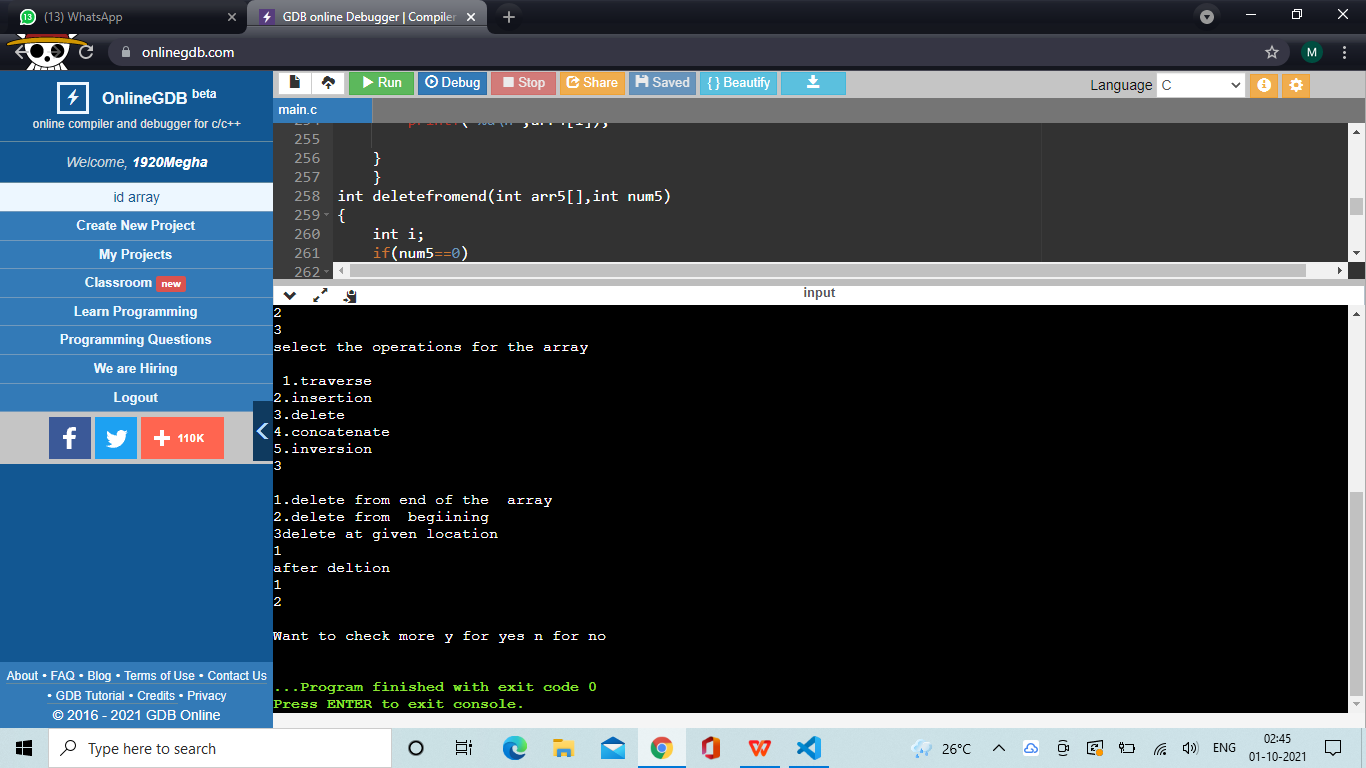
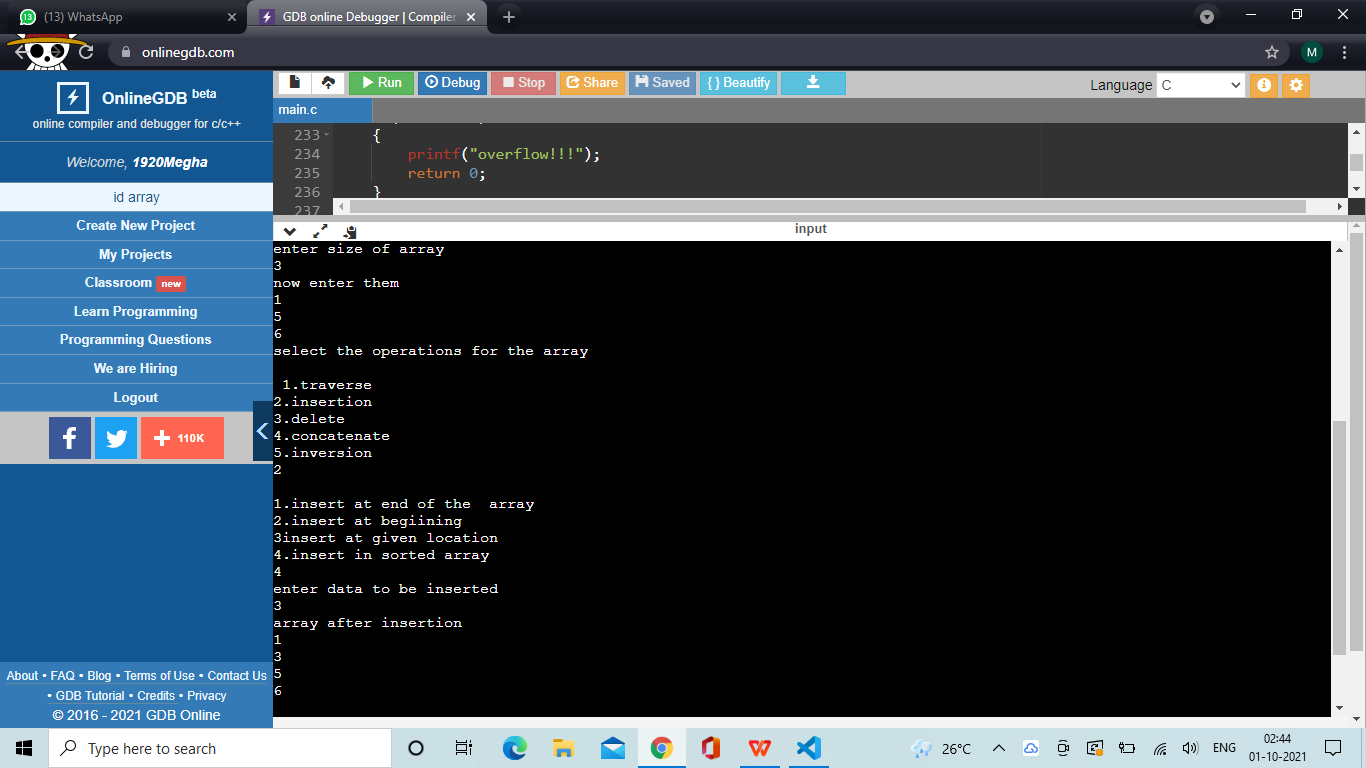
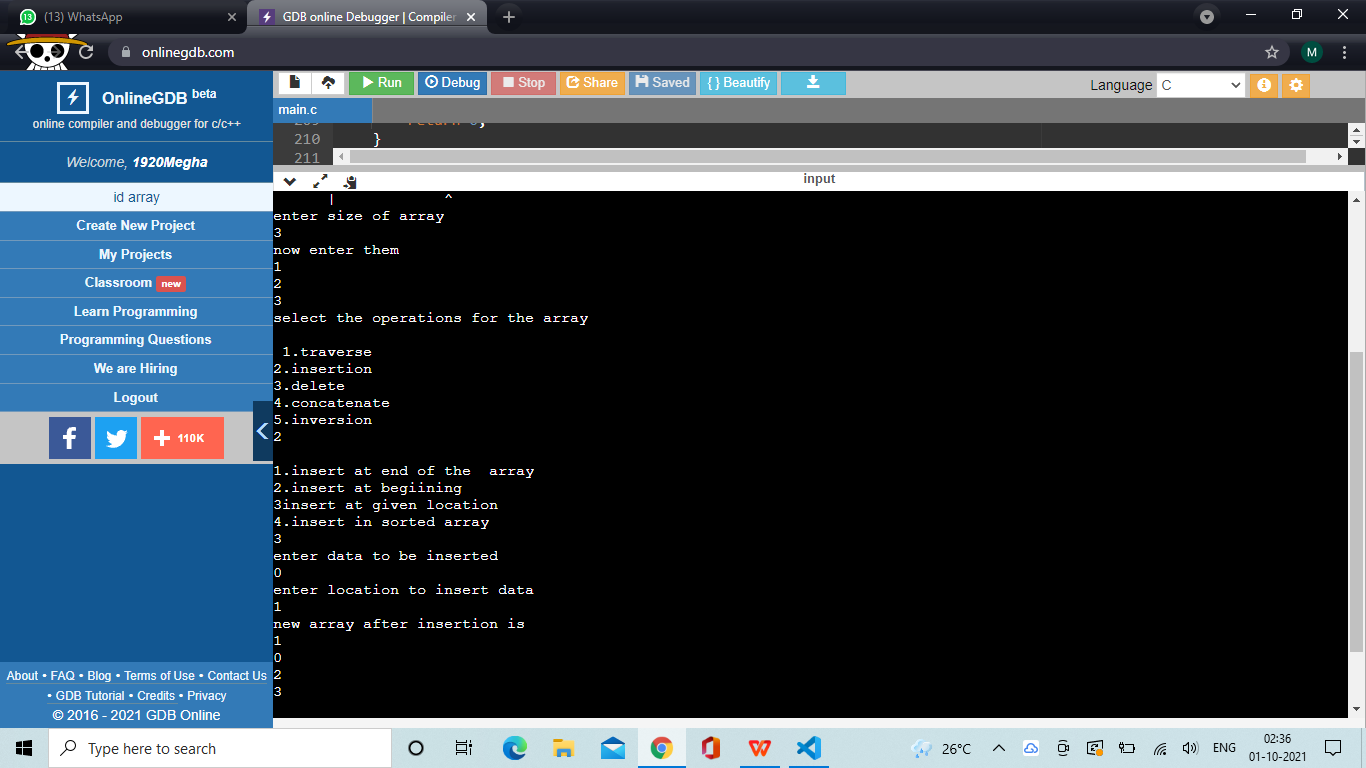
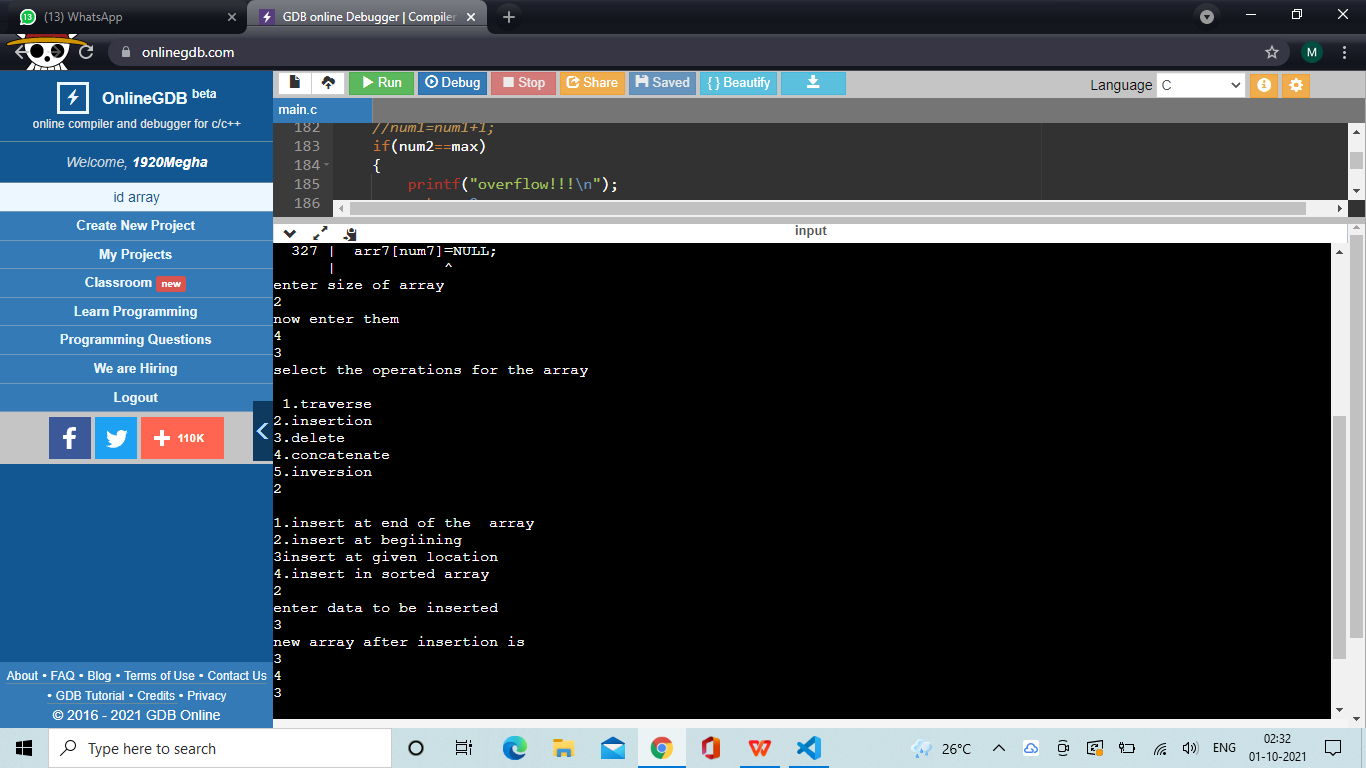
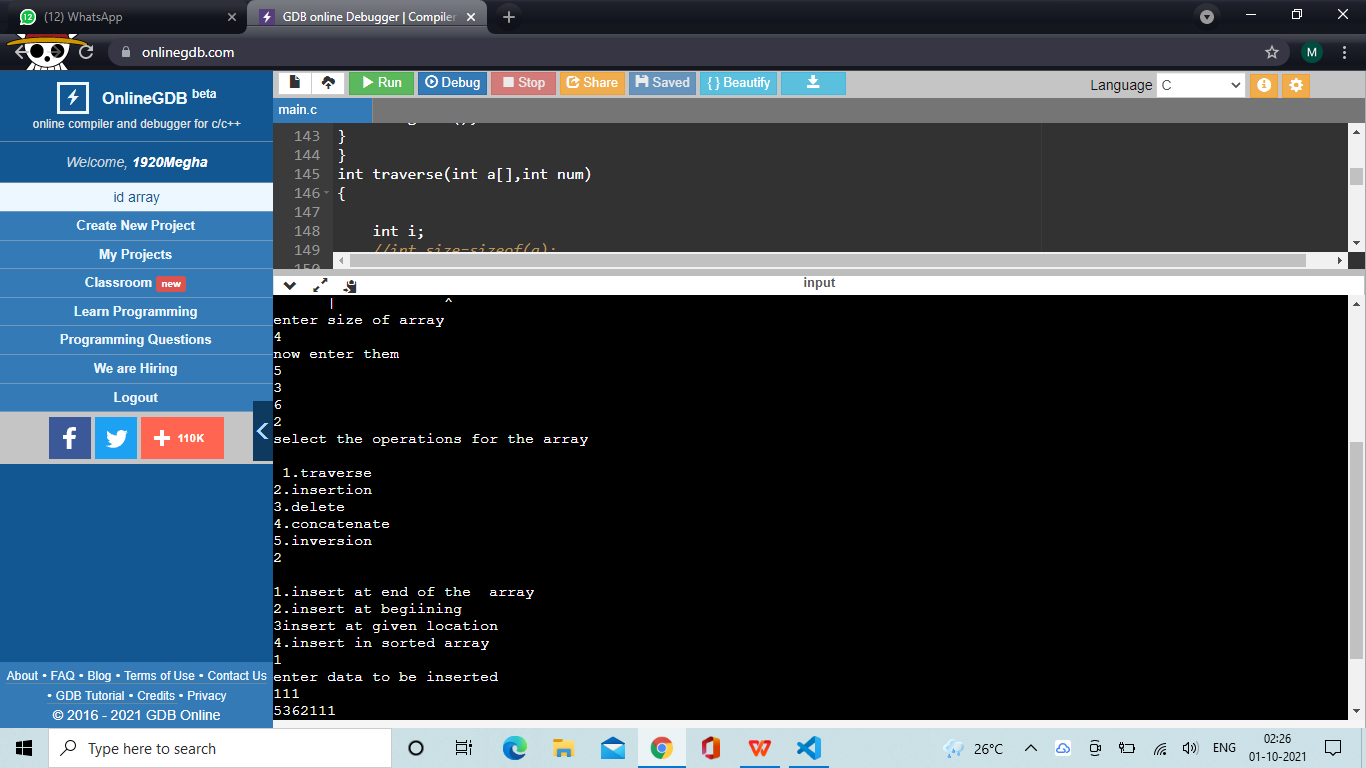
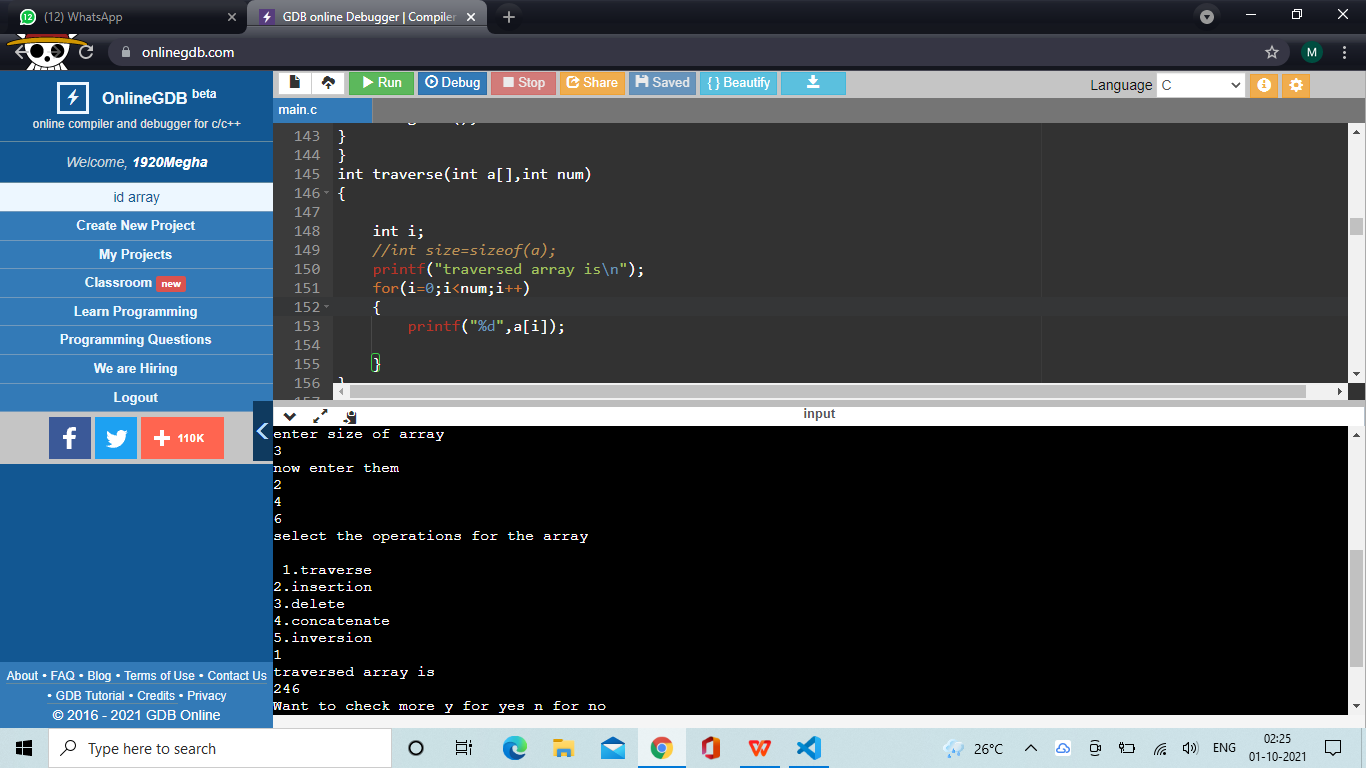
for(k=1;k<num11+1;k++)

{

printf("%d ",arr11[k]);

}

}



**Code to merge two sorted into 3rd sorted array**

#include<stdio.h>

void main(){

int c[30],k,j,ch1,a[10],b[10],n1,n2,i;

printf("enter size of array1\n");

scanf("%d",&n1);

printf("now enter them in SORTED WAY\n");

for(i=0;i<n1;i++)

{

scanf("%d",&a[i]);

}

printf("enter size of array2\n");

scanf("%d",&n2);

printf("now enter them in SORTED WAY\n");

for(i=0;i<n2;i++)

{

scanf("%d",&b[i]);

}

int size=n1+n2;

i=0;j=0;k=0;

//while(i<n1 && j<n2)

for(k=0;k<size;k++){

if(a[i]<b[j] && i!=n1){

c[k]=a[i];

i++;

// k++;

}

else{

c[k]=b[j];

j++;

//k++;

}

}

if(n1<n2)

{

while(j<n2){

c[k-1]=b[j];

j++;

k++;}

}

else{

while(i<n1)

{c[k-1]=a[i];

i++;

k++;}

}

printf("printing merged array in sorted order\n");

for(k=0;k<size;k++)

{

printf("%d ",c[k]);

}

}

