1. **Check whether the given number is Ugly or not**

**#include<stdio.h>**

**#include<math.h>**

**int main(){**

**int n=300;**

**int j=0;**

**j=isugly(n);**

**if(j==1)**

**printf("ugly %d",n);**

**else**

**printf("not ugly %d",n);**

**return 0;**

**}**

**int isugly(int n){**

**n=check(n,2);**

**n=check(n,3);**

**n=check(n,5);**

**return n;**

**}**

**int check(int a,int b){**

**while(a>1&&(a%b==0)){**

**a=a/b;**

**}**

**return a;**

**}**

1. **Sorting based on given weights**

**#include<stdio.h>**

**#include<math.h>**

**int main(){**

**//5-perfect square,4-multiple of 4 and divisible by 6, 3-even**

**int array1[5]={25,54,89,12,33};**

**int array2[5]={};**

**int n,temp=0,size=sizeof(array1)/sizeof(array1[0]);**

**float b;**

**for(int i=0;i<size;i++){**

**b=(float)sqrt(array1[i]);**

**n=b;**

**if(n==b){**

**array2[i]=5;**

**}**

**else if((array1[i]%4==0)||(array1[i]%6==0)){**

**array2[i]=4;**

**}**

**else if(array1[i]%2==0){**

**array2[i]==3;**

**}**

**else{**

**array2[i]=1;**

**}**

**}**

**for(int i=0;i<size;i++){**

**for(int j=0;j<size;j++){**

**if(array1[i]>array1[j]){**

**temp=array1[i];**

**array1[i]=array1[j];**

**array1[j]=temp;**

**temp=array2[i];**

**array2[i]=array2[j];**

**array2[j]=temp;**

**}**

**}**

**}**

**for(int i=0;i<size;i++){**

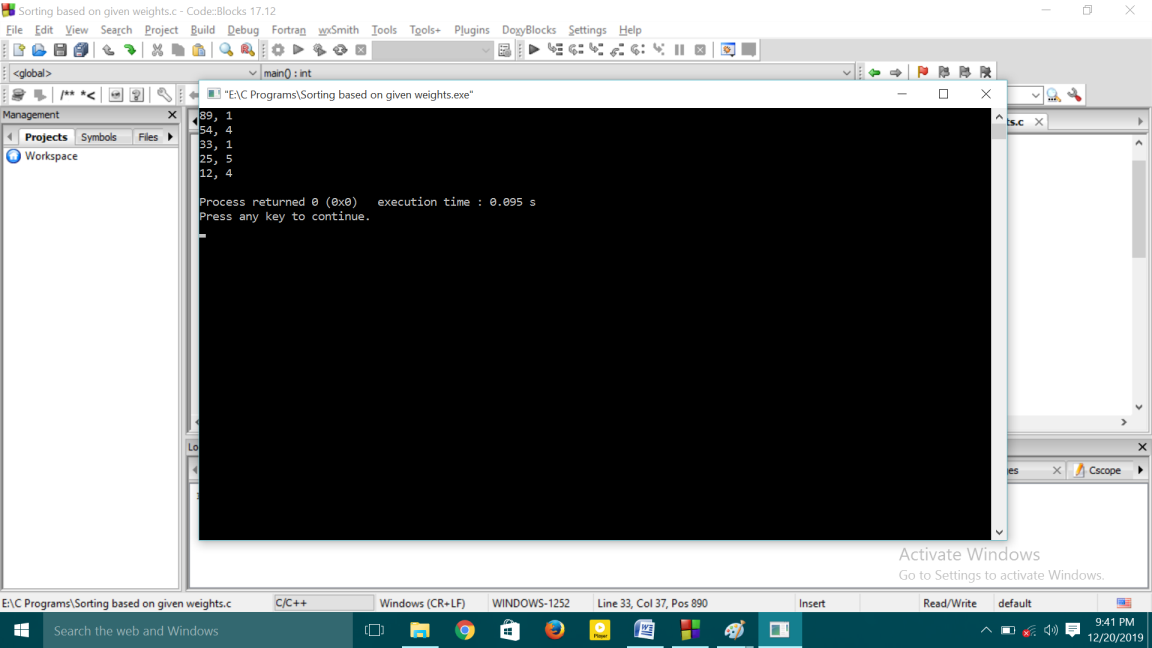
**printf("%d, %d\n",array1[i],array2[i]);**

**}**

**return 0;**

**}**

**o/p:**

****

1. **Sort the elements based on the number of factors**

**#include<stdio.h>**

**int main(){**

**int arr[5]={4,26,58,6,50};**

**int arr2[5]={},temp=0;**

**for(int i=0;i<5;i++){**

**printf("%d ",arr[i]);**

**}**

**printf("\n\n");**

**for(int i=0;i<5;i++){**

**int count=0;**

**for(int j=1;j<=arr[i];j++){**

**if(arr[i]%j==0){**

**count++;**

**}**

**arr2[i]=count;**

**}**

**}**

**for(int i=0;i<5;i++){**

**printf("%d->%d\n\n",arr[i],arr2[i]);**

**}**

**printf("\n\n");**

**for(int i=0;i<4;i++){**

**for(int j=i+1;j<5;j++){**

**if(arr2[j]>=arr2[i]){**

**temp=arr[i];**

**arr[i]=arr[j];**

**arr[j]=temp;**

**temp=arr2[i];**

**arr2[i]=arr2[j];**

**arr2[j]=temp;**

**}**

**}**

**}**

**for(int i=0;i<5;i++){**

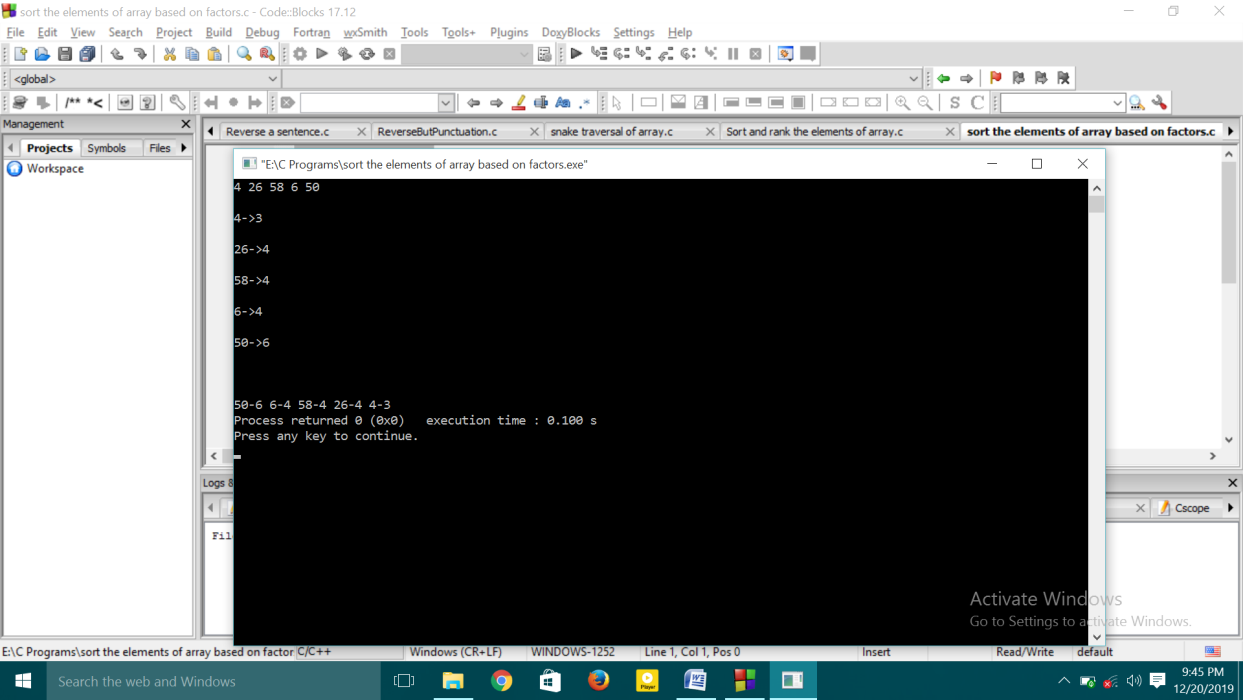
**printf("%d-%d ",arr[i],arr2[i]);**

**}**

**return 0;**

**}**

**o/p:**

****

1. **Sort and Rank the elements of the array**

**#include<stdio.h>**

**#include<math.h>**

**int main(){**

**int arr[10]={10,20,30,30,40,40,50,50,60,70};**

**int n=sizeof(arr)/sizeof(arr[0]);**

**int arr2[10]={};**

**int temp=0,count=0,rank=1;**

**for(int i=0;i<n;i++){**

**for(int j=0;j<n;j++){**

**if(arr[i]<arr[j]){**

**temp=arr[i];**

**arr[i]=arr[j];**

**arr[j]=temp;**

**}**

**}**

**}**

**for(int i=0;i<n;i++){**

**count=1;**

**while(arr[i]==arr[i+1]){**

**count++;**

**i++;**

**}**

**arr2[i]=count;**

**}**

**for(int i=0;i<n;i++){**

**if(arr2[i]==0){**

**rank++;**

**}**

**else{**

**if(arr[i]!=arr[i+1]){**

**printf("%d-%d\n",arr[i],rank);**

**rank++;**

**}**

**else{**

**rank++;**

**}**

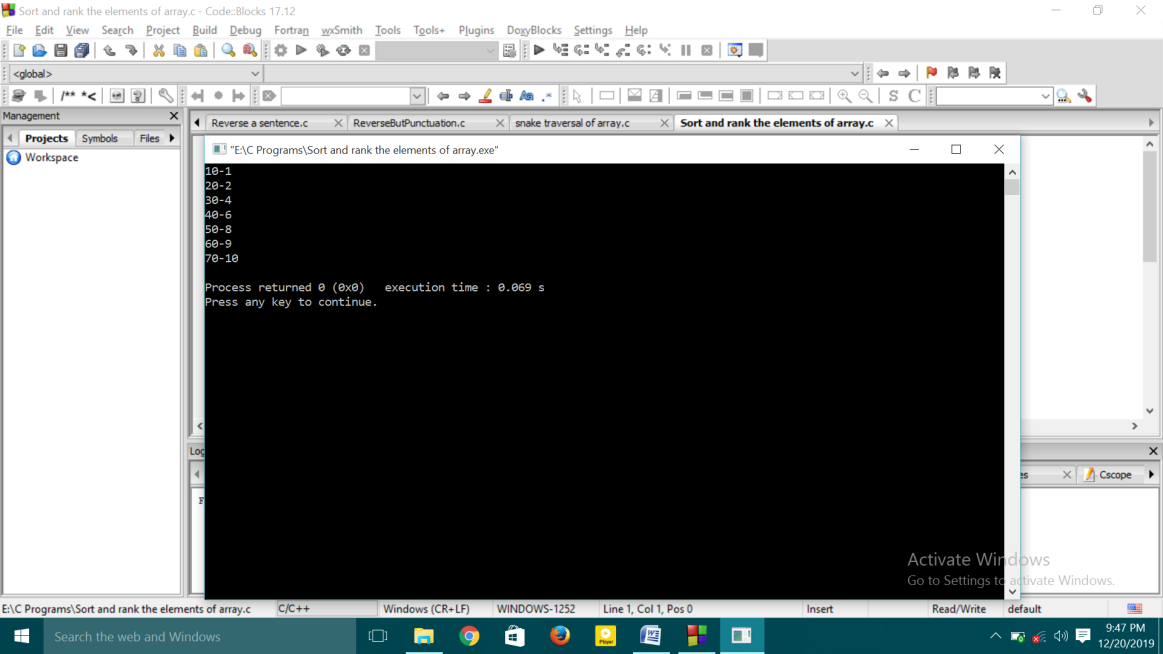
**}**

**}**

**return 0;**

**}**

**o/p:**

****

1. **Snake Traversal of an array**

**#include<stdio.h>**

**int main(){**

**int arr[4][3]={{1,2,3},{4,5,6},{7,8,9},{10,11,12}};**

**for(int i=0;i<4;i++){**

**for(int j=0;j<3;j++){**

**printf("%d ",arr[i][j]);**

**}**

**printf("\n");**

**}**

**for(int i=0;i<4;i++){**

**for(int j=0,k=2;j<3;j++,k--){**

**if(i%2==0){**

**printf("%d ",arr[i][j]);**

**}**

**else{**

**printf("%d ",arr[i][k]);**

**}**

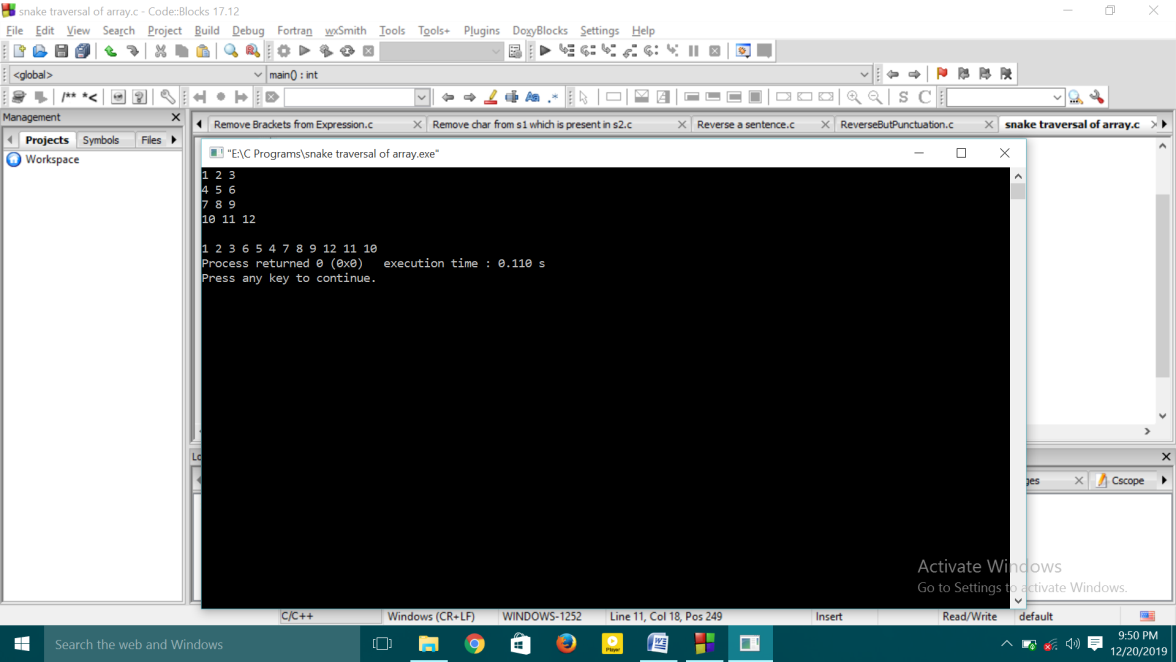
**}**

**}**

**return 0;**

**}**

**o/p:**

****

1. **Reverse the sentence leaving the punctuations**

**#include<stdio.h>**

**#include<string.h>**

**char \*reverseFunction(char str[]){**

**int i=0,j=strlen(str)-1;**

**char t;**

**while(i<j){**

**while((str[i]<65 || str[i]>90) && (str[i]<97 || str[i]>122))**

**i++;**

**while((str[j]<65 || str[j]>90) && (str[j]<97 || str[j]>122))**

**j--;**

**t=str[j];**

**str[j]=str[i];**

**str[i]=t;**

**i++;j--;**

**while((str[i]<65 || str[i]>90) && (str[i]<97 || str[i]>122))**

**i++;**

**while((str[j]<65 || str[j]>90) && (str[j]<97 || str[j]>122))**

**j--;**

**}**

**return (char \*)str;**

**}**

**void main(){**

**char str[100];**

**gets(str);**

**printf("%s",reverseFunction(str));**

**}**

1. **Reverse a Sentence**

**#include<stdio.h>**

**#include<string.h>**

**int main(){**

**char sen[]="One Two Three four five";**

**int n=strlen(sen);**

**int n2=n;**

**printf("%s\n",sen);**

**for(int i=n-1;i>=0;i--){**

**if(sen[i]==' '||sen[i]=='\0'){**

**for(int j=i+1;j<=n2;j++){**

**printf("%c",sen[j]);**

**}**

**sen[i]='\0';**

**n2=i;**

**}**

**}**

**printf("%s",sen);**

**return 0;**

**}**

1. **Remove the Characters from one string present in other string**

**#include<stdio.h>**

**#include<string.h>**

**int main(){**

**char s1[5]="Hello";**

**char s2[2]="ell";**

**int flag;**

**printf("%s\n",s1);**

**for(int i=0;i<5;i++){**

**flag=0;**

**for(int j=0;j<2;j++){**

**if(s1[i]==s2[j]){**

**flag=1;**

**break;**

**}**

**}**

**if(flag==1)**

**printf("%c",s1[i]);**

**}**

**return 0;**

**}**

1. **Remove brackets from the given expression**

**#include<stdio.h>**

**#include<string.h>**

**#define totsize 50**

**char stack[totsize];**

**int top=-1;**

**void push(char a){**

**if(top==(totsize-1))**

**printf("Stack Overflow");**

**else**

**top++;**

**stack[top]=a;**

**}**

**void pop(){**

**if(top<0)**

**printf("Stack is empty");**

**else**

**top--;**

**}**

**int main(){**

**char actualexpr[20]="a-(b-(c+d))+e";**

**char newstring[10]="";**

**int flag=1,idx2=0;**

**push('+');**

**for(int i=0;i<strlen(actualexpr);i++){**

**printf("%c",actualexpr[i]);**

**}**

**for(int i=0;i<strlen(actualexpr);i++){**

**if(actualexpr[i]=='+'){**

**if(stack[top]=='-'){**

**newstring[idx2]='-';**

**idx2++;**

**}**

**else{**

**newstring[idx2]='+';**

**idx2++;**

**}**

**}**

**else if(actualexpr[i]=='-'){**

**if(stack[top]=='-'){**

**newstring[idx2]='+';**

**idx2++;**

**}**

**else{**

**newstring[idx2]='-';**

**idx2++;**

**}**

**}**

**else if(actualexpr[i]=='(' && i!=0){**

**if(actualexpr[i-1]=='-')**

**push('-');**

**else**

**push('+');**

**}**

**else if(actualexpr[i]==')'){**

**pop();**

**}**

**else{**

**newstring[idx2]=actualexpr[i];**

**idx2++;**

**}**

**}**

**printf("\n");**

**for(int i=0;i<strlen(newstring);i++){**

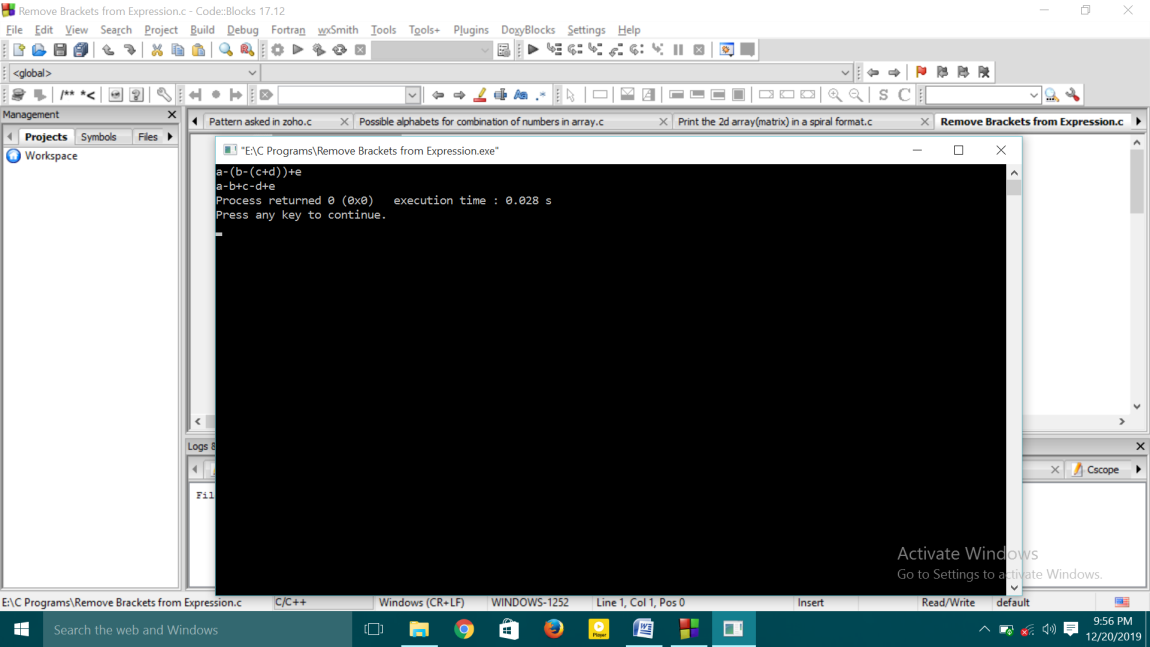
**printf("%c",newstring[i]);**

**}**

**return 0;**

**}**

**o/p:**

****

1. **Spiral format of 2d array**

**#include<stdio.h>**

**#include<conio.h>**

**int main(){**

**int arr[4][3]={{1,2,3},{4,5,6},{7,8,9},{10,11,12}};**

**printf("Given Array:\n");**

**for(int i=0;i<4;i++){**

**for(int j=0;j<3;j++){**

**printf("%d ",arr[i][j]);**

**}**

**printf("\n");**

**}**

**printf("\n");**

**rotatearray(arr,4,3);**

**return 0;**

**}**

**void rotatearray(int arr[4][3],int r,int c){**

**int rindex=0,cindex=0,rend=r-1,cend=c-1;**

**while(rindex<=rend&&cindex<=cend){**

**for(int i=rindex;i<=cend;i++){**

**printf("%d ",arr[rindex][i]);**

**}**

**rindex++;**

**for(int i=rindex;i<=rend;i++){**

**printf("%d ",arr[i][cend]);**

**//printf("(%d,%d): %d",i,cend-1,arr[cend-1][i]);**

**}**

**cend--;**

**//printf("%d,%d %d\n\n",cend,cindex,rend);**

**for(int i=cend;i>=cindex;i--){**

**printf("%d ",arr[rend][i]);**

**//printf("(%d,%d): %d",rend,i,arr[rend][i]);**

**}**

**rend--;**

**for(int i=rend;i>cindex;i--){**

**printf("%d ",arr[i][cindex]);**

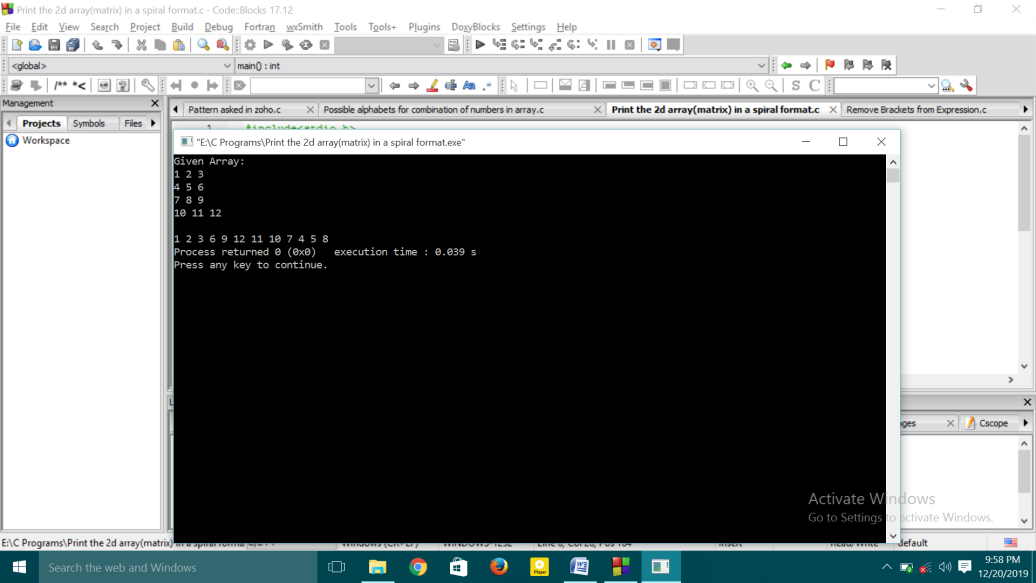
**}**

**cindex++;**

**}**

**}**

**o/p:**

****

1. **Possible alphabet combination of numbers in the array**

**#include<stdio.h>**

**#include<math.h>**

**int main()**

**{**

**int arr[5]= {1,1,2,4,5},arr2[5]={};**

**int a=64,temp=0,val=0,j,k=0;**

**printf("1 1 2 4 5\n");**

**for(int i=0; i<5; i++)**

**{**

**printf("%c ",(a+arr[i]));**

**}**

**printf("\n");**

**for(int i=0; i<5; i++)**

**{**

**int flag=1;**

**for(j=0,k=0; j<5; j++,k++)**

**{**

**val=arr[j];**

**if(j==i)**

**{**

**val=(arr[j]\*10)+arr[j+1];**

**j++;**

**}**

**if(val<27){**

**arr2[k]=(a+val);**

**}**

**else{**

**flag=0;**

**break;**

**}**

**}**

**if(flag){**

**for(int z=0;z<k;z++)**

**printf("%c ",arr2[z]);**

**}**

**printf("\n");**

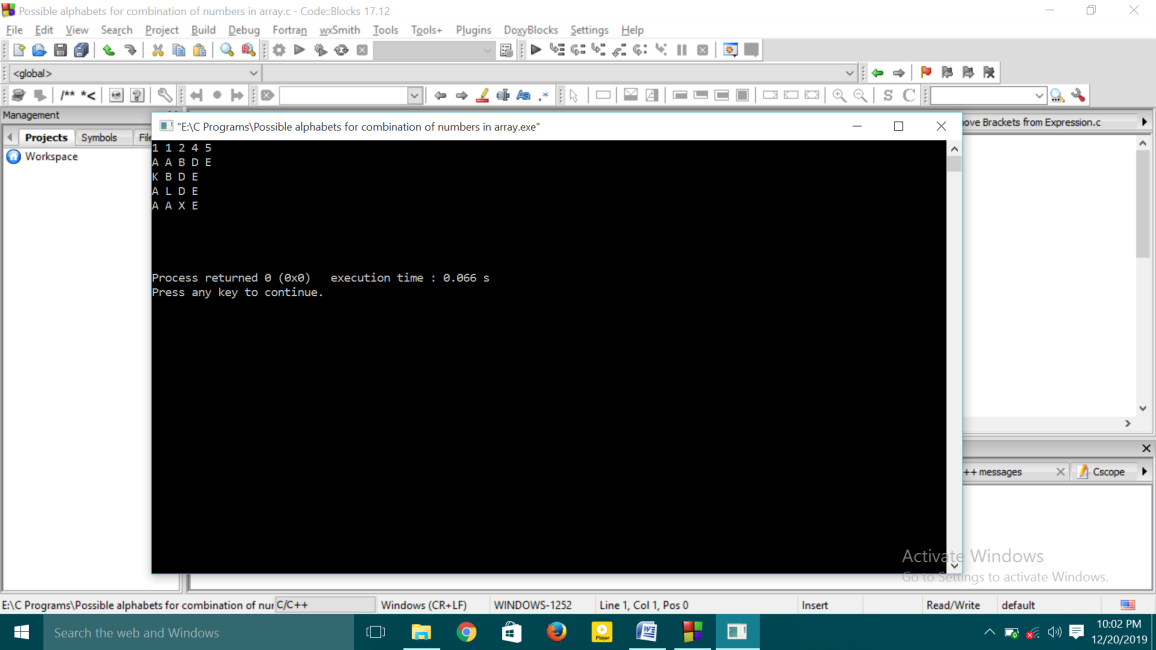
**}**

**printf("\n");**

**return 0;**

**}**

**o/p:**

****

1. **X-Pattern**

**#include<stdio.h>**

**int main(){**

**int n=5;**

**for(int i=1;i<=n/2;i++){**

**for(int j=1;j<=i;j++){**

**if(j%2==0){**

**printf("0");**

**}**

**else{**

**printf("1");**

**}**

**}**

**for(int k=1;k<=n-(2\*i);k++){**

**printf("-");**

**}**

**for(int j=1;j<=i;j++){**

**if(j%2==0){**

**printf("0");**

**}**

**else{**

**printf("1");**

**}**

**}**

**printf("\n");**

**}**

**for(int j=0;j<n;j++){**

**if(j%2==0){**

**printf("1");**

**}**

**else{**

**printf("0");**

**}**

**}**

**printf("\n");**

**for(int i=n/2;i>=1;i--){**

**for(int j=1;j<=i;j++){**

**if(j%2==0){**

**printf("0");**

**}**

**else{**

**printf("1");**

**}**

**}**

**for(int k=n-(2\*i);k>=1;k--){**

**printf("-");**

**}**

**for(int j=1;j<=i;j++){**

**if(j%2==0){**

**printf("0");**

**}**

**else{**

**printf("1");**

**}**

**}**

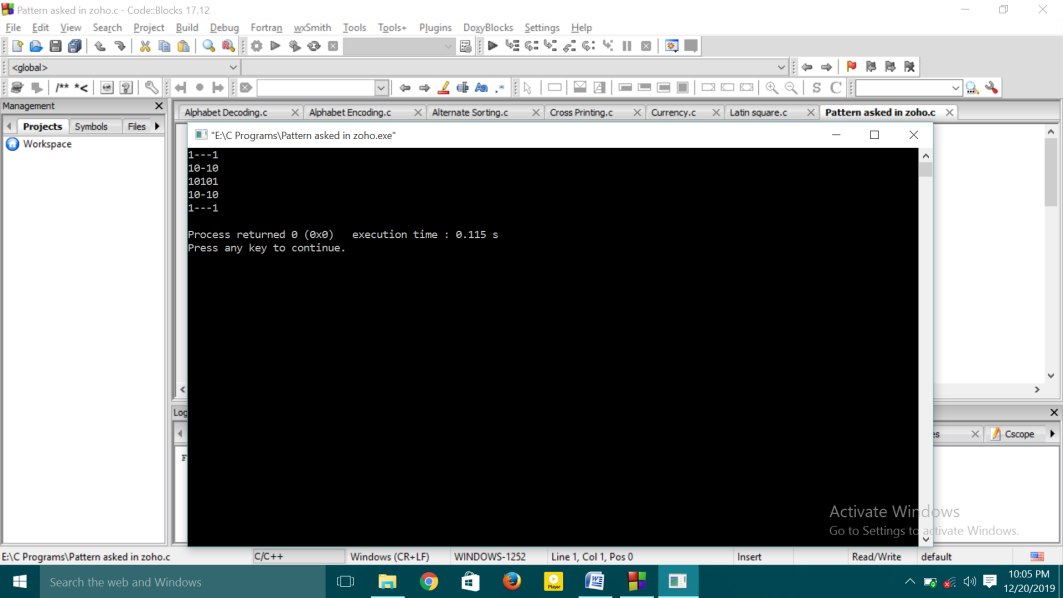
**printf("\n");**

**}**

**return 0;**

**}**

**o/p:**

****

1. **Latin Square**

**#include<stdio.h>**

**int main(){**

**int n=5,temp=0;**

**int k=n;**

**for(int i=0;i<n;i++){**

**temp=n;**

**for(int j=k;j<n;j++){**

**printf("%d ",j);**

**}**

**while(temp>0){**

**printf("%d ",temp);**

**temp--;**

**if(temp==i)**

**break;**

**}**

**printf("\n");**

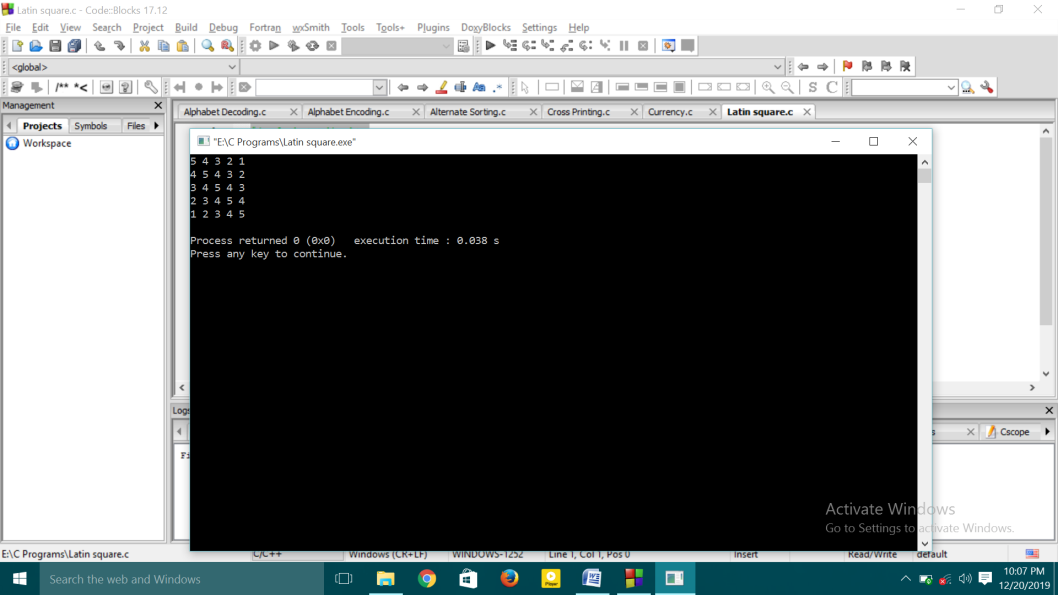
**k--;**

**}**

**return 0;**

**}**

**o/p:**

****

1. **Currency Denomination**

**#include<stdio.h>>**

**#include<math.h>**

**int main(){**

**int amount=10;**

**int curr[6]={2000,500,100,50,20,10};**

**int count[6]={};**

**for(int i=0;i<6;i++){**

**if(amount>=curr[i]){**

**count[i]=amount/curr[i];**

**amount=amount-(curr[i]\*count[i]);**

**}**

**}**

**for(int i=0;i<6;i++){**

**printf("%d: %d\n",curr[i],count[i]);**

**}**

**return 0;**

**}**

1. **Alternate Sorting**

**#include<stdio.h>**

**#include<math.h>**

**int main(){**

**int n=0,arr[6]={9,6,7,2,1,4};**

**n=sizeof(arr)/sizeof(arr[0]);**

**int temp=0,first=0,last=n-1,fin[6]={},flag=0;**

**for(int i=0;i<n;i++){**

**for(int j=0;j<n;j++){**

**if(arr[i]>arr[j]){**

**temp=arr[i];**

**arr[i]=arr[j];**

**arr[j]=temp;**

**}**

**}**

**}**

**for(int i=0;i<n;i++){**

**if(flag==0){**

**fin[i]=arr[first];**

**first++;**

**flag=1;**

**}**

**else{**

**fin[i]=arr[last];**

**last--;**

**flag=0;**

**}**

**}**

**for(int i=0;i<n;i++){**

**printf("%d ",arr[i]);**

**}**

**printf("\n");**

**for(int i=0;i<n;i++){**

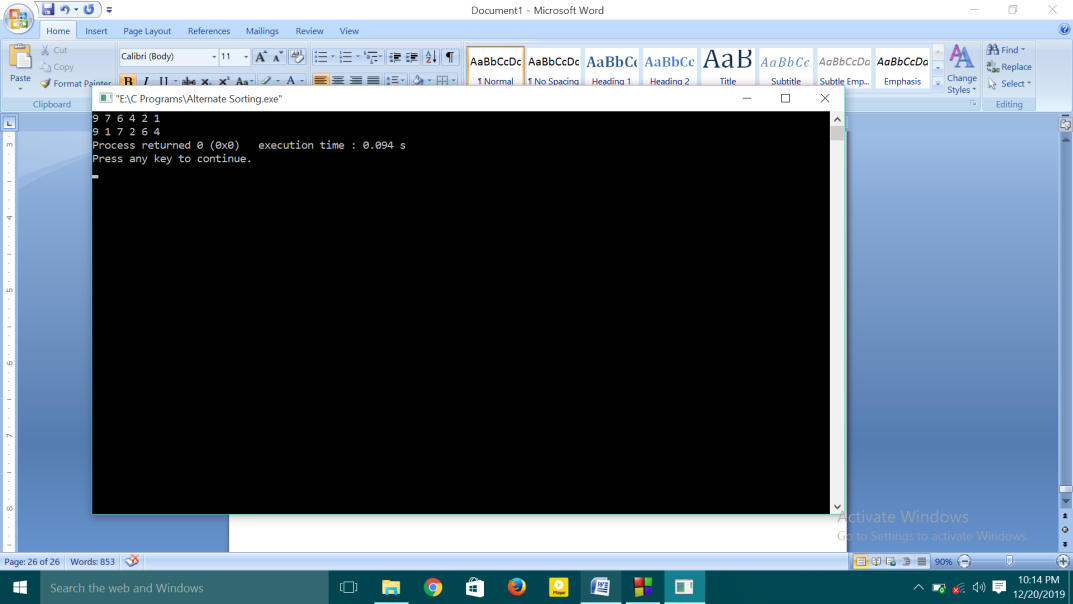
**printf("%d ",fin[i]);**

**}**

**return 0;**

**}**

**O/p:**

****