# Q1. Sort string in alphabetical order

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//bubble sort

void sort(char str[], int n)

{

    int i,j;

    for(i=0;i<n-1;++i){

        //after each loop the swapping term decreases by 1

        for(j=0;j<n-i-1;++j){

            if(str[j]>str[j+1]){

                swap(&str[j],&str[j+1]);

            }

        }

    }

}

void swap(char\* a, char\* b){

    char temp = \*a;

    \*a=\*b;

    \*b=temp;

}

int main()

{

    char str[30];

    printf("Enter a string ...\n");

    gets(str);

    int n=strlen(str);

    sort(str,n);

    printf("The sorted string is ...\n");

    int i;

    for(i=0;i<n;++i){

        //if statement to not to print the spaces

        if(str[i]!=' '){

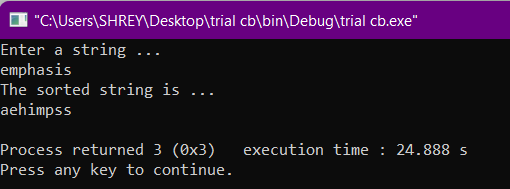
            printf("%c",str[i]);

        }

    }

    printf("\n");

    return 3;

}

# Q2.Print an input number in words. E.g. 42 = forty two

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

    int num;

    printf("Enter a number : ");

label\_2:

    scanf("%d",&num);

    int temp=num;

    int temp\_2=num;

    int digit=0;

    while(temp!=0){

        temp=temp/10;

        ++digit;

    }

    if(num==0) printf("Zero");

    //\*\*IMPORTANT PART IS HOW TO INITIALIZE STRING INSIDE ARRAY\*\*//

    char \*ones[]={"Zero","one","two","three","four","five","six","seven","eight","nine"};

    char \*two[]={"","eleven","twelve","thirteen","fourteen","fifteen","sixteen","seventeen","eighteen","nineteen"};

    char \*tens[]={"","","twenty","thirty","forty","fifty","sixty","seventy","eighty","ninety"};

    switch (digit)

    {

        case 1:

            if(num!=0){

                //\*\*HAVE A LOOK AT THE PRINTING SYNTAX\*\*//

                //HERE \*ones[num] IS NOT USED//

                printf("%s\n",ones[num]);

            }

            break;

        case 2:

        label:

            if(num<20){

                printf("%s",two[num%10]);

            }

            else if(num%10==0){

                printf("%s",tens[num/10]);

            }

            else{

                printf("%s %s",tens[num/10],ones[num%10]);

            }

            //reinitializing value of num

            num=temp\_2;

            break;

        case 3:

            if(num!=100){

                printf("%s hundred ",ones[num/100]);

                //value of num is being changed

                num=num%100;

                goto label;

            }

            else printf("One Hundred");

            break;

        default:

            printf("Please enter a valid input...\n");

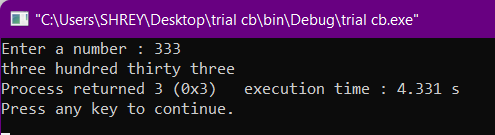
            goto label\_2;

            break;

    }

    return 3;

}



# Q3. Delete n characters from a given position in a string. E.g. deleting “as” from “assassination” gives “ssination”.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

    int i,j;

    printf("Enter the string...\n");

    char str[30];

    char \*ptr=str;

    gets(str);

    int m=strlen(str);

    char del[20];

    printf("Enter a substring to be removed : ");

    gets(del);

    int n=strlen(del);

    for(i=0;i<m-1;++i){

        int count=0;

        for(j=0;j<n;++j){

            if(\*(ptr+i+j)==del[j]){

                ++count;

            }

        }

        if(count==n){

            int temp;

            for(int k=i;k<m;k++){

                \*(ptr+k)=\*(ptr+k+n);

            }

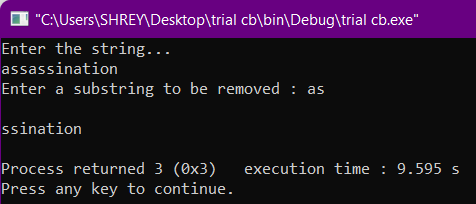
        }

    }

    printf("\n%s\n",str);

    return 3;

}



# Q4. Insert a substring after every occurrence of a character. E.g. inserting “as” after every “e” in “evermore” gives “easveasrmoreas”.

#include <stdio.h>

#include <stdlib.h>

int main()

{

    printf("Enter a string --> ");

    char str[30];

    char \*ptr=str;

    gets(str);

    printf("Enter the character after which you want to insert : ");

    char after;

    scanf("%c",&after);

    printf("Enter the substring to be inserted --> ");

    char ins[10];

    fflush(stdin);

    gets(ins);

    int i=0;

    while(\*(ptr+i)!='\0'){

        if(\*(ptr+i)==after){

            printf("%c%s",after,ins);

        }

        else printf("%c",\*(ptr+i));

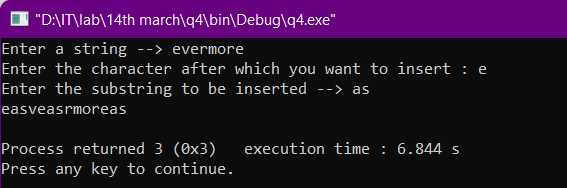
        ++i;

    }

    printf("\n");

    return 3;

}



# Q5. Print the distance travelled at each second using given value of acceleration and computing distance and velocity repeatedly.

#include <stdio.h>

#include <stdlib.h>

void dist(float u,float a,int t)

{

    if(t!=0){

        float x2 = (u\*t) + (1/2.0)\*a\*t\*t;

        float x1 = (u\*(t-1)) + (1/2.0)\*a\*(t-1)\*(t-1);

        printf("d = %.3f, ",x2-x1);

        printf("Total distance covered : %.3f\n",x2);

    }

    else{

        float x = (u\*t) + (1/2.0)\*a\*t\*t;

        printf("d = %.3f, ",x);

        printf("Total distance covered : %.3f\n",x);

    }

}

void vel(float u,float a,int t)

{

    u = u + a\*(t);

    printf("u = %.3f, ",u);

}

int main()

{

    int time;

    printf("Enter the time duration (in sec) : ");

    scanf("%d",&time);

    float velocity;

    printf("Enter the value of initial velocity (in m/s) : ");

    scanf("%f",&velocity);

    float accelaration;

    printf("Enter the value of accelaration (in m/s^2) : ");

    scanf("%f",&accelaration);

    printf("\n\n");

    float distance;

    float \*ptr=&distance;

    float \*ptr\_2=&velocity;

    for(int i=0;i<time;++i){

        printf("At time t = %dsec, ",i);

        vel(velocity,accelaration,i);

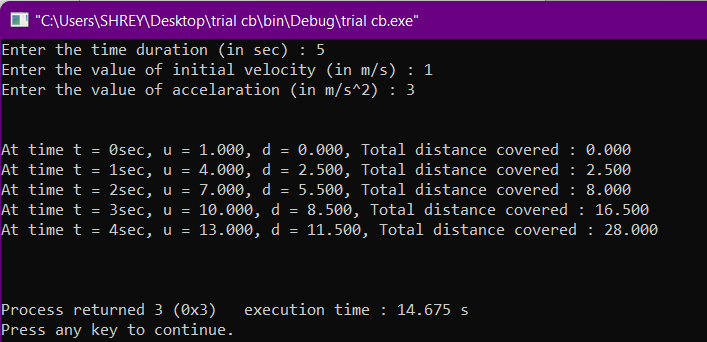
        dist(velocity,accelaration,i);

    }

    printf("\n\n");

    return 3;

}



# Q6. Generate a pyramid of \* symbols (asterisk).

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int i,j,k;

    printf("Enter the no of rows : ");

    int rows;

    scanf("%d",&rows);

    for(i=1;i<=rows;++i){

        //before space

        for(j=0;j<rows-i;++j){

            printf(" ");

        }

        for(k=0;k<i;++k){

            printf("\* ");

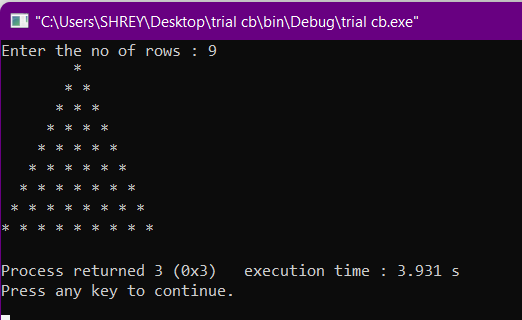
        }

        printf("\n");

    }

    return 3;

}



# Q7. Generate a diamond of @ symbols(at).

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int i,j,k;

    //here rows represents the upper part of the diamond

    printf("Enter the no of rows : ");

    int rows;

    scanf("%d",&rows);

    for(i=0;i<rows;++i){

        //printing the spaces

        for(j=0;j<rows-i;++j){

            printf(" ");

        }

        for(k=0;k<i+1;++k){

            printf("@ ");

        }

        printf("\n");

    }

    //printing the middle part

    for(j=0;j<rows+1;++j){

        printf("@ ");

    }

    printf("\n");

    //printing the lower part

    for(k=0;k<rows;++k){

        //printing the spaces

        for(i=0;i<k+1;++i){

            printf(" ");

        }

        for(j=0;j<rows-k;++j){

            printf("@ ");

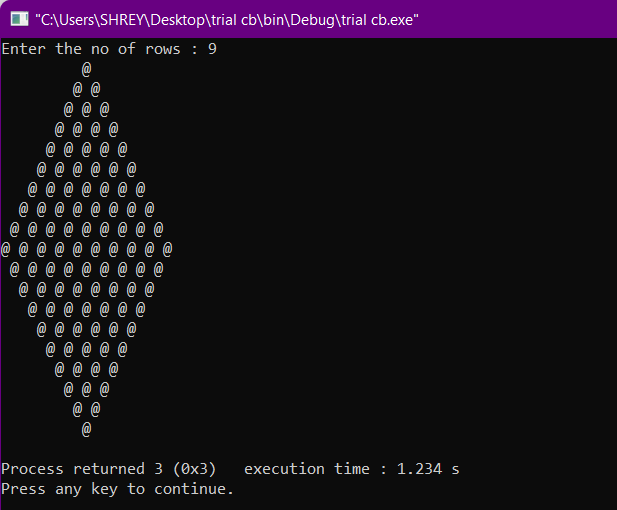
        }

        printf("\n");

    }

    return 3;

}



# Q8.Find divisibility of a number by 2,3,4,5,6,8,9,10 and 11 using divisibility rule.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

    int i,j,k;

    int div\_2=0,div\_3=0,div\_4=0,div\_5=0,div\_6=0,div\_8=0,div\_9=0,div\_10=0,div\_11=0;

    int num;

    printf("Enter the number : ");

    scanf("%d",&num);

    int temp=num;

    int sum=0;

    int rem;

    while(temp!=0){

        rem=temp%10;

        temp=temp/10;

        sum+=rem;

    }

    if(num%2==0){

        div\_2 = 1;

        printf("The number is divisible by 2\n");

    }

    if(sum%3==0){

        div\_3=1;

        printf("The number is divisible by 3\n");

    }

    /\*last two digits must be divisible by 4\*/

    if((num%100)%4==0){

        div\_4=1;

        printf("The number is divisible by 4\n");

    }

    if(num%5==0){

        div\_5=1;

        printf("The number is divisible by 5\n");

    }

    if((div\_2==1)&&(div\_3==1)){

        div\_6=1;

        printf("The number is divisible by 6\n");

    }

    /\*last 3 digits must be divisible by 8\*/

    if((num%1000)%8==0){

        div\_8=1;

        printf("The number is divisible by 8\n");

    }

    if(sum%9==0){

        div\_9=1;

        printf("The number is divisible by 9\n");

    }

    if((div\_2==1)&&(div\_5==1)){

        div\_10=1;

        printf("The number is divisible by 10\n");

    }

    int no=0;

    sum=0;

    while(num!=0){

        ++no;

        rem=num%10;

        num=num/10;

        sum+=pow(-1,no)\*rem;

    }

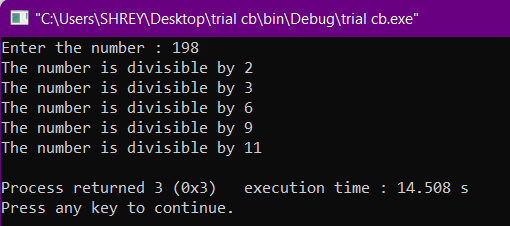
    if(sum==0||sum%11==0){

        printf("The number is divisible by 11\n");

    }

    return 3;

}



# Q9. Iteratively find and print the topper student in student list as long as user is giving inputs. Use dynamic memory allocation.

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

int main()

{

    int i=1;

    float \*marks = (float\*)malloc(1\*sizeof(int));

    float max=0;

    while(1){

        printf("Student\_%d marks : ",i);

        scanf("%f",&\*marks);

        if(\*marks>max){

            max=\*marks;

            printf("The topper is Student\_%d with marks : %.3f",i,max);

        }

        else printf("The topper is Student\_%d with marks : %.3f",i-1,max);

        free(marks);

        float \*marks = (float\*)malloc(1\*sizeof(int));

        ++i;

        printf("\n\nPress 0 to exit... ");

        int input;

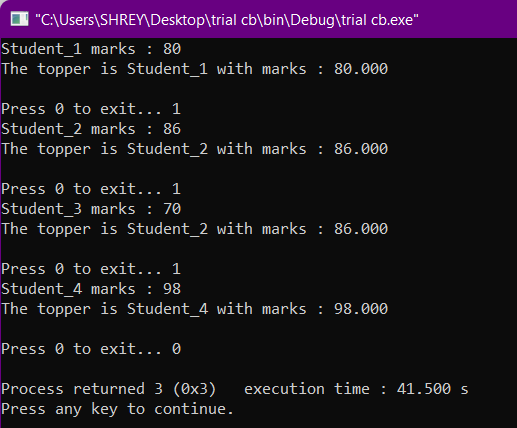
        scanf("%d",&input);

        if(input==0) break;

    }

    return 3;

}



# Q10. Find the mean,median and mode of a randomly generated array of integers. Use pointers.

#include <stdio.h>

#include <stdlib.h>

void swap(int \*a,int \*b)

{

    int temp=\*a;

    \*a=\*b;

    \*b=temp;

}

int main()

{

    int \*arr = (int\*)calloc(1,sizeof(int));

    printf("Enter the value inside the array...\n\n");

    int input=-1;

    int i=0;

    int sum=0;

    //creating a freq ptr for storing frequency

    // int \*freq = (int\*)calloc(1,sizeof(int));

    //taking input and reallocationg the array till the user press '0'

    while(input!=0){

        printf("Number %d : ",i+1);

        scanf("%d",&\*(arr+i));

        arr = (int\*)realloc(arr,(i+2)\*sizeof(int));

        //taking sum till the last input

        sum+=\*(arr+i);

        ++i;

        printf("Press 0 to exit ... ");

        scanf("%d",&input);

        printf("\n");

    }

    //printing the mean value

    printf("\nThe mean is : %.3f\n",sum/(float)i);

    float mean=sum/(float)i;

    int median;

    //printing the median value

    if(i%2!=0){

        int position = i/2;

        printf("The median is : %d",\*(arr+position));

        median=\*(arr+position);

    }

    else{

        int position = i/2;

        printf("The median is : %.3f",(\*(arr+position)+\*(arr+position+1))/2.0);

        median=(\*(arr+position)+\*(arr+position+1))/2.0;

    }

    //mode=mean-3\*(mean-median)

    printf("\nThe mode is : %.3f",mean-3\*(mean-median));

    return 3;

}

