**Important Programs**

**String Functions**

1)strlen()

#include<stdio.h>

#include<string.h>

#include "cassert"

int my\_strlen(const char\*);

int main(){

int n=0;

const char \*str="Jyoti";

n=my\_strlen(str);

printf("The length of the string \"%s\" is %d\n",str,n);

return 0;

}

int my\_strlen(const char \*lptr){

assert (lptr);

int length=0;

while(\*lptr++){

length++;

}

return length;

}

2) strcpy ()

#include<stdio.h>

#include <stdlib.h>

#include "cassert"

char\* my\_strcpy(char\*,const char\*);

int main(){

const char source[] = "Jyoti";

char destination[40]="Ranjan";

my\_strcpy(destination,source);

printf("The Destination is \"%s\" \n",destination);

return 0;

}

char\* my\_strcpy(char \*dest,const char \*src){

assert (dest && src);

char\* temp=dest;

while(\*dest++ = \*src++);

return temp;

}

3) strcat ()

#include<stdio.h>

#include <stdlib.h>

#include "cassert"

char\* my\_strcat(char\*,const char\*);

int main(){

const char source[] = "ranjan";

char destination[40]="Jyoti";

my\_strcat(destination,source);

printf("The Destination is \"%s\" \n",destination);

return 0;

}

char\* my\_strcat(char \*dest,const char \*src){

assert (dest && src);

char\* temp=dest;

if(\*dest!=0){

while(\*++dest);

}

while(\*dest++ = \*src++);

return temp;

}

4) strcmp ()

#include<stdio.h>

#include <stdlib.h>

#include "cassert"

int my\_strcmp(const char\*,const char\*);

int main(){

int n=0;

const char source[] = "J";

const char destination[]="j";

n=my\_strcmp(destination,source);

printf("N is \"%d\" \n",n);

return 0;

}

int my\_strcmp(const char \*s1, const char \*s2) {

assert (s1 && s2);

while (\*s1 && \*s2) {

if (\*s1 == \*s2) {

++s1;

++s2;

}

else {

break;

}

}

return \*s1 - \*s2;

}

4) atoi ()

#include <stdio.h>

#include <stdlib.h>

int my\_atoi(const char\*);

int main()

{

const char \*str = "1123.56";

int n = my\_atoi(str);

//int n = atoi(str);

printf("The string %s as an integer is = %d\n",str,n);

return 0;

}

int my\_atoi(const char \*lptr){

int result=0;

while(1){

if ((\*lptr >= '0') && (\*lptr <= '9')){

result \*= 10;

result += \*lptr - '0';

lptr++;

}

else{

break;

}

}

return result;

}

Swap two variables using macro

#include<stdio.h>

#include<conio.h>

#define SWAPE(x,y) int t;t=x;x=y;y=t;

int main()

{

int a,b;

printf("\n Enter two numbers");

scanf("%d%d",&a,&b);

printf("\n Before swaping the Value of a=%d and b=%d",a,b);

SWAPE(a,b);

printf("\n After swap value of a=%d and b=%d",a,b);

return 0;

}

**Multiply by 5 with given number using bitwise operator**

#include<stdio.h>

#include<conio.h>

int main(){

int a;

printf("\n Enter One number");

scanf("%d",&a);

printf("\n After multiply by 5, the number is %d",(a<<2)+a);

return 0;

}

**REVERSE A NUMBER USING ARITHMATIC OPERATOR**

#include <stdio.h>

int main() {

int num,rev=0;

printf("Enter a number:");

scanf("%d", &num);

while(num>0){

rev=(rev\*10)+ (num%10);

num=num/10;

}

printf("Reverse of the given number: %d", rev);

return 0;

}

**PRINT THE GIVEN COMMAND LINE ARGUMENTS**

#include<stdio.h>

int main(int argc,char \*\*argv){

int i;

printf("argc=%d\n",argc);

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

printf("argv[%d] = %s\n",i,argv[i]);

return 0;

}

**Write a line backward**

#include<stdio.h>

void wrt\_it();

int main(){

printf("Type a line ...");

wrt\_it();

return 0;

}

void wrt\_it(){

int c;

if((c=getchar())!='\n')

wrt\_it();

putchar(c);

}

**Print 12345678910987654321**

#include<stdio.h>

int main(){

int i=1,j=1;

while(i<20){

if(i<=10){

printf("%d",i);

j=i-1;

i++;

}else{

printf("%d",j);

j--;

i++;

}

}

return 0;

}

**Variable Length Arguments**

#include<stdio.h>

#include<stdarg.h>

void sum(char \*, int, ...);

int main(void){

sum("The sum of 10+15+13 is %d.\n",3,10,15,13);

return 0;

}

void sum(char \*string,int num\_args,...){

int sum=0;

va\_list ap;

int loop;

va\_start(ap,num\_args);

for(loop=0;loop<num\_args;loop++)

sum+=va\_arg(ap,int);

printf(string,sum);

va\_end(ap);

}

**Variable Length Arguments**

#include <stdio.h>

#include <stdarg.h>

void printargs(int arg1, ...) /\* print all int type args, finishing with -1 \*/

{

va\_list ap;

int i;

va\_start(ap, arg1);

for (i = arg1; i != -1; i = va\_arg(ap, int))

printf("%d ", i);

va\_end(ap);

putchar('\n');

}

int main(void)

{

printargs(5, 2, 14, 84, 97, 15, 24, 48, -1);

printargs(84, 51, -1);

printargs(-1);

printargs(1, -1);

return 0;

}

**swap two numbers without using temp variable**

#include<stdio.h>

int main(){

int a=5;

int b=10;

a=a^b;

b=a^b;

a=a^b;

printf("a=%d",a);

printf("b=%d",b);

return 0;

}

**function to dynamically allocate 2-dimensional array using malloc.**

void allocate2D(int\*\* arr, int nrows, int ncols) {

arr = (int\*\*)malloc(nrows\*sizeof(int\*));

for(int i=0;i<nrows;i++){

arr[i]=(int\*)malloc(ncols\*sizeof(int));

}

}

**Palindrome Check**

#include<stdio.h>

#include<string.h>

int check\_Palindrome(char\*);

int main(){

char name[1024];

printf("Enter a string for Palindrome Check \n");

scanf("%s",name);

if(check\_Palindrome(name))

printf("This is a Palindrome\n");

else

printf("This is not a Palindrome\n");

return 0;

}

int check\_Palindrome(char\* str){

int num=(int)strlen(str);

for(int i=0;i<num/2;i++){

if(str[i]!=str[num-(i+1)]){

return 0;

}

}

return 1;

}

**Count no of ‘1’ in a given integer**

#include <stdio.h>

int main (void){

int var = 7;

int i, count = 0;

while(var > 0)

{

count += var & 1;

var >>= 1;

}

printf("%d\n", count);

return 0;

}

**Without Using “sizeof” operator print size of given structure**

#include<stdio.h>

struct MyStruct

{

int i;

int j;

char a;

};

int main()

{

struct MyStruct \*p=0;

int size = ((char\*)(p+1))-((char\*)p);

printf("\nSIZE : %d\n", size);

return 0;

}

**Arrenge 0’s and 1’s in given array.**

#include<stdio.h>

int main(){

int i,a[10]={0};

int \*r,\*w;

a[2]=a[4]=a[6]=a[8]=1;

for(i=0;i<10;i++)

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

r=w=a;

for(i=0;i<10;i++){

if(\*r==1){

\*w=1;

\*r=0;

++w;

}

++r;

}

printf("\n\nafter\n");

for(i=0;i<10;i++)

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

return 0;

}

**Factorila Of a Number Using Recursion**

#include<stdio.h>

int my\_factorial(int);

int main(){

printf("%d",my\_factorial(10));

return 0;

}

int my\_factorial(int a){

if(a <=1){

return 1;

}else{

return (a\*my\_factorial(a-1));

}

}