

1 . Program to find Fibonacci numbers

```
#include<stdio.h>
int main()
{
int n1 = 0;
int n2 = 1;
int n3 = 0;
int i = 0;
int number = 0;
printf("Enter the number of fibonacci to be printed");
scanf("%d",&number);
printf("\n %d\n %d\n",n1,n2);
for(i = 2; i < number; ++i)
{
n3 = n1+n2;
printf(" %d \n",n3);
n1 = n2;
n2 = n3;
}
return 0;
}
```

2. Check if a number is prime or not

```
#include<stdio.h>
int main()
{
int n;
int i;
int m=0;
int flag=0;
printf("Enter number to check prime or not\n");
scanf("%d",&n);
m = n/2;
for(i = 2; i <= m;i++)
{
if(n%i == 0)
{
printf("Number is NOT prime\n");
return 0;
}
}
if(flag==0)
printf("Number is prime\n");
return 0;
}
```

```
}
```

3. Check if number is palindrome or not

```
#include<stdio.h>
int main()
{
    int n;
    int r;
    int sum=0;
    int temp;
    printf("Enter number to check palindrome or not\n");
    scanf("%d",&n);
    temp = n;
    while(n>0)
    {
        r = n%10;
        sum = (sum*10)+r;
        n = n/10;
    }
    if(temp == sum)
        printf("Number is palindrome \n");
    else
        printf("Number is NOT palindrome\n");
    return 0;
}
```

4. Find factorial of the number

```
#include<stdio.h>
int main()
{
    int i;
    int fact=1;
    int number;
    printf("Enter a number: \n");
    scanf("%d",&number);
    for(i = 1; i <= number; i++)
    {
        fact=fact*i;
    }
    printf("Factorial of %d is: %d\n",number,fact);
    return 0;
}
```

5. Check if a number is Armstrong number

```
#include<stdio.h>
int main()
{
    int n;
    int r;
    int sum=0;
    int temp;
    printf("Enter a number \n");
    scanf("%d",&n);
    temp=n;
    while(n>0)
    {
        r=n%10;
        sum=sum+(r*r*r);
        n=n/10;
    }
    if(temp == sum)
        printf("Number is a armstrong number\n");
    else
        printf("Number is NOT a armstrong number\n");
    return 0;
}
```

6. Program to find sum of digits

```
#include<stdio.h>
int main()
{
    int n;
    int sum=0;
    int m;
    printf("Enter number:\n");
    scanf("%d",&n);
    while(n>0)
    {
        m = n%10;
        sum = sum+m;
        n = n/10;
    }
    printf("Sum is = %d\n",sum);
    return 0;
}
```

7. Reverse a number

```
#include<stdio.h>
int main()
{
    int n;
    int reverse = 0;
    int rem;
    printf("Enter a number: \n");
    scanf("%d", &n);
    while(n!=0)
    {
        rem = n%10;
        reverse = reverse*10+rem;
        n /= 10;
    }
    printf("Reversed Number: %d\n",reverse);
    return 0;
}
```

8. Swap Numbers in C

```
#include<stdio.h>
int main()
{
    int a = 10;
    int b = 20;
    printf("Before swap a=%d b=%d\n",a,b);
    a = a+b;
    b = a-b;
    a = a-b;
    printf("\nAfter swap a=%d b=%d\n",a,b);
    return 0;
}
```

10. Print hello world without printf

```
#include<stdio.h>
int main()
{
    if(printf("hello world\n")){}
    return 0;
}
```

10. Program without main

```
.
#include<stdio.h>
#define start main
void start()
{
printf("Hello\n");
}
```

11. Program for decimal to binary

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int a[10];
int n;
int i;
printf("Enter decimal number \n");
scanf("%d",&n);
for(i=0;n>0;i++)
{
a[i]=n%2;
n=n/2;
}
printf("\nBinary number is ");
for(i=i-1;i>=0;i--)
{
printf("%d",a[i]);
}
return 0;
}
```

12. Check if number is positive or negative

```
#include <stdio.h>
void main()
{
int num;
printf("Enter a number: \n");
scanf("%d", &num);
if (num > 0)
printf("%d is a positive number \n", num);
}
```

```

else if (num < 0)
printf("%d is a negative number \n", num);
}

```

13. Program to find greatest of 3 numbers

```

#include<stdio.h>
int main()
{
int num1,num2,num3;
printf("\nEnter value of num1, num2 and num3:\n");
scanf("%d %d %d",&num1,&num2,&num3);
if((num1>num2)&&(num1>num3))
printf("\n Number1 is greatest\n");
else if((num2>num3)&&(num2>num1))
printf("\n Number2 is greatest\n");
else
printf("\n Number3 is greatest\n");
return 0;
}

```

14. Program to get ascii value

```

#include <stdio.h>
int main()
{
char ch;
printf("Enter character:\n");
scanf("%c", &ch);
printf("ASCII value of character %c is: %d\n", ch, ch);
return 0;
}

```

15. Program to demonstrate sizeof function

```

#include<stdio.h>
int main()
{
printf("Size of char: %ld byte\n",sizeof(char));
printf("Size of int: %ld bytes\n",sizeof(int));
printf("Size of float: %ld bytes\n",sizeof(float));
printf("Size of double: %ld bytes\n", sizeof(double));
}

```

```
return 0;
}
```

16. Program to check for leap year

```
#include <stdio.h>
int main()
{
    int y;
    printf("Enter year: \n");
    scanf("%d",&y);
    if(y % 4 == 0)
    {
        if( y % 100 == 0)
        {
            if ( y % 400 == 0)
            printf("%d is a Leap Year\n", y);
            else
            printf("%d is not a Leap Year\n", y);
        }
        else
        printf("%d is a Leap Year\n", y );
    }
    else
    printf("%d is not a Leap Year\n", y);
    return 0;
}
```

17. Program to convert upper case to lower case

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[25];
    int i;
    printf("Enter the string: \n");
    scanf("%s",str);
    for(i=0;i<=strlen(str);i++)
    {
        if(str[i]>=65&&str[i]<=90)
        str[i]=str[i]+32;
    }
}
```

```
printf("\nLower Case String is: %s\n",str);
return 0;
}
```

18. Program to convert lower case to upper case

```
#include<stdio.h>
#include<string.h>
int main()
{
char str[25];
int i;
printf("Enter the string:\n");
scanf("%s",str);
for(i=0;i<=strlen(str);i++)
{
if(str[i]>=97&&str[i]<=122)
str[i]=str[i]-32;
}
printf("\nUpper Case String is: %s\n",str);
return 0;
}
```

19. Program to get string length

```
#include <stdio.h>
int main()
{
char str[100],i;
printf("Enter a string: \n");
scanf("%s",str);
for(i=0; str[i]!='\0'; ++i);
printf("\nLength of input string: %d",i);
return 0;
}
```

20. Program to concatenate 2 strings

```
#include <stdio.h>
int main()
{
char str1[50], str2[50], i, j;
printf("\nEnter first string: ");
scanf("%s",str1);
```



```

printf("\nEnter second string: ");
scanf("%s",str2);
for(i=0; str1[i]!='\0'; ++i);
for(j=0; str2[j]!='\0'; ++j, ++i)
{
    str1[i]=str2[j];
}
str1[i]='\0';
printf("\nOutput: %s\n",str1);
return 0;
}

```

21. Program to get largest element in an array

```

#include <stdio.h>
int largest_element(int arr[], int num)
{
    int i, max_element;
    max_element = arr[0];
    for (i = 1; i < num; i++)
        if (arr[i] > max_element)
            max_element = arr[i];
    return max_element;
}
int main()
{
    int arr[] = {1, 24, 15, 20, 8, -11, 30};
    int n = sizeof(arr)/sizeof(arr[0]);
    printf("Largest element of array is %d", largest_element(arr, n));
    return 0;
}

```

22. Program to get size of the array

```

#include <stdio.h>
int main()
{
    double arr[] = {1, 2, 3, 4, 5, 6};
    int n;
    n = sizeof(arr) / sizeof(arr[0]);
    printf("Size of the array is: %d\n", n);
    return 0;
}

```

23. Program to get GCD

```
#include <stdio.h>
int main()
{
    int n1, n2, i, gcd;
    printf("Enter two integers: \n");
    scanf("%d %d", &n1, &n2);
    for(i=1; i <= n1 && i <= n2; ++i)
    {
        if(n1%i==0 && n2%i==0)
            gcd = i;
    }
    printf("G.C.D of %d and %d is %d\n", n1, n2, gcd);
    return 0;
}
```

24. Program to get LCM

```
#include <stdio.h>
int main()
{
    int n1, n2, minMultiple;
    printf("Enter two positive integers: \n");
    scanf("%d %d", &n1, &n2);
    minMultiple = (n1>n2) ? n1 : n2;
    while(1)
    {
        if( minMultiple%n1==0 && minMultiple%n2==0 )
        {
            printf("The LCM of %d and %d is %d.\n", n1, n2,minMultiple);
            break;
        }
        ++minMultiple;
    }
    return 0;
}
```

25. Program to get number of digits

```
#include <stdio.h>
int main()
{
    long long n;
    int count = 0;
```

```
printf("Enter an integer: ");
scanf("%lld", &n);
while(n != 0)
{
    n /= 10;
    ++count;
}
printf("Number of digits: %d", count);
}
```

26. Program to get factorial

```
#include <stdio.h>
int main()
{
    int number, i;
    printf("Enter a positive integer: ");
    scanf("%d",&number);
    printf("Factors of %d are: ", number);
    for(i=1; i <= number; ++i)
    {
        if (number%i == 0)
        {
            printf("%d ",i);
        }
    }
    return 0;
}
```

27. Program to get array average

```
#include <stdio.h>
int main()
{
    int n, i;
    float num[100], sum = 0.0, average;
    printf("Enter the numbers of elements: ");
    scanf("%d", &n);
    while (n > 100 || n <= 0)
    {
        printf("Error! number should in range of (1 to 100).\n");
        printf("Enter the number again: ");
        scanf("%d", &n);
    }
    for(i = 0; i < n; ++i)
```

```

{
printf("%d. Enter number: ", i+1);
scanf("%f", &num[i]);
sum += num[i];
}
average = sum / n;
printf("Average = %.2f", average);
return 0;
}

```

28. Program for strcpy

```

#include <stdio.h>
int main()
{
char s1[100], s2[100], i;
printf("Enter string s1: ");
scanf("%s",s1);
for(i = 0; s1[i] != '\0'; ++i)
{
s2[i] = s1[i];
}
s2[i] = '\0';
printf("String s2: %s", s2);
return 0;
}

```

29. Program for strrev

```

#include<stdio.h>
#include<string.h>
int main()
{
char name[30] = "Hello";
printf("String before strrev( ) : %s\n",name);
printf("String after strrev( ) : %s",strrev(name));
return 0;
}

```

30. Program to check strong number

```

#include<stdio.h>
int main()
{
int num,i,fact,r,sum=0,temp;
printf("Please enter a number to find strong number");

```

```

scanf("%d",&num);
temp=num;
while(num)
{
i=1,fact=1;
r=num%10;
while(i<=r)
{
fact=fact*i;
i++;
}
sum=sum+fact;
num=num/10;
}

if(sum==temp)
printf("\nThe number %d is a strong number",temp);
else
printf("\nThe number %d is not a strong number",temp);
return 0;
}

```

31. Program to get number cube

```

#include<stdio.h>
int main()
{
int n;
printf("Enter a number");
scanf("%d",&n);
printf("\nSquare of the number %d is %d",n,n*n);
printf("\nCube of the number %d is %d",n,n*n*n);
return 0;
}

```

32. Program to generate random number

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
int i, n;
printf("Five random numbers between 1 and 100000\n");
for (i = 1; i <= 5; i++)
{

```

```

n = rand()%100000 + 1;
printf("%d\n", n);
}
return 0;
}

```

33. Program to check if strings are anagram

```

#include <stdio.h>
#include <string.h>
int main (void) {
char s1[] = "recitals";
char s2[] = "articles";
char temp;
int i, j;
int n = strlen(s1);
int n1 = strlen(s2);
// If both strings are of different length, then they are not anagrams
if( n != n1) {
printf("%s and %s are not anagrams! \n", s1, s2);
return 0;
}
// lets sort both strings first -
for (i = 0; i < n-1; i++) {
for (j = i+1; j < n; j++) {
if (s1[i] > s1[j]) {
temp = s1[i];
s1[i] = s1[j];
s1[j] = temp;
}
if (s2[i] > s2[j]) {
temp = s2[i];
s2[i] = s2[j];
s2[j] = temp;
}
}
}
// Compare both strings character by character

for(i = 0; i<n; i++) {
if(s1[i] != s2[i]) {
printf("Strings are not anagrams! \n", s1, s2);
return 0;
}
}
}

```

```
printf("Strings are anagrams! \n");
return 0;
}
```

34. Program for strcpy

```
#include <stdio.h>
int main()
{
char s1[] = "prodevelopertutorial";
char s2[8];
int length = 0;
while(s1[length] != '\0')
{
s2[length] = s1[length];
length++;
}
s2[length] = '\0';
printf("Value in s1 = %s \n", s1);
printf("Value in s2 = %s \n", s2);
return 0;
}
```

35. Program to toggle case

```
#include <stdio.h>
#define MAX_SIZE 100
void toggleCase(char * str)
{
int i = 0;
while(str[i] != '\0')
{
if(str[i]>='a' && str[i]<='z')
{
str[i] = str[i] - 32;
}
else if(str[i]>='A' && str[i]<='Z')
{
str[i] = str[i] + 32;
}
i++;
}
}
int main()
{
```

```

char str[MAX_SIZE];
/* Input string from user */
printf("Enter any string: ");
gets(str);
printf("String before toggling case: %s", str);
toggleCase(str);
printf("String after toggling case: %s", str);
return 0;
}

```

36. Program to get number of vowels and consonant

```

#include <stdio.h>
#include <string.h>
#define MAX_SIZE 100 // Maximum string size
int main()
{
char str[MAX_SIZE];
int i, len, vowel, consonant;
/* Input string from user */
printf("Enter any string: ");
gets(str);
vowel = 0;
consonant = 0;
len = strlen(str);
for(i=0; i<len; i++)
{
if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z'))
{
/*
* If the current character(str[i]) is a vowel
*/
if(str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||
str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U' )
vowel++;
else
consonant++;
}
}
printf("Total number of vowel = %d\n", vowel);
printf("Total number of consonant = %d\n", consonant);
return 0;
}

```


37. Program to get nth bit of a number

```
#include <stdio.h>
int main()
{
    int num, n, bitStatus;
    printf("Enter any number: ");
    scanf("%d", &num);
    printf("Enter nth bit to check (0-31): ");
    scanf("%d", &n);
    bitStatus = (num >> n) & 1;
    printf("The %d bit is set to %d", n, bitStatus);
    return 0;
}
```

38. Program to set nth bit of a number

```
#include <stdio.h>
int main()
{
    int num, n, newNum;
    printf("Enter any number: ");
    scanf("%d", &num);
    printf("Enter nth bit to set (0-31): ");
    scanf("%d", &n);
    newNum = (1 << n) | num;
    printf("Bit set successfully.\n\n");
    printf("Number before setting %d bit: %d (in decimal)\n", n, num);
    printf("Number after setting %d bit: %d (in decimal)\n", n, newNum);
    return 0;
}
```

39. Program to toggle bits of a number

```
#include <stdio.h>
int main()
{
    int num, n, newNum;
    printf("Enter any number: ");
    scanf("%d", &num);
    printf("Enter nth bit to toggle (0-31): ");
    scanf("%d", &n);
    newNum = num ^ (1 << n);
    printf("Number before toggling %d bit: %d (in decimal)\n", n, num);
    printf("Number after toggling %d bit: %d (in decimal)\n", n, newNum);
    return 0;
}
```

40. Program to get even or odd

```
#include <stdio.h>
int main()
{
    int num;
    printf("Enter any number: ");
    scanf("%d", &num);
    if(num & 1)
    {
        printf("%d is odd.", num);
    }
    else
    {
        printf("%d is even.", num);
    }
    return 0;
}
```

41. Program for sigalarm

```
.
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
void sig_alarm(int signo)
{
    if(signo==SIGALRM) {
        printf("Got SIGALRM\n");
        alarm(10);
        printf("%s: Will generate alarm in 10 secs\n",__FUNCTION__);
    } else {
        printf("Unknown signal number\n");
    }
}
int main(void)
{
    struct sigaction act;
    act.sa_handler = sig_alarm;
    act.sa_flags = 0;
    sigemptyset(&act.sa_mask);
    if (sigaction (SIGALRM, &act, 0) == -1) {
        perror (NULL);
        return 1;
    }
    alarm (10);
    printf ("%s: Will generate alarm in 10 secs\n",__FUNCTION__);
```

```
while(1) pause();
return 0;
}
```

42. Program to get asctime

```
#include <stdio.h>
#include <time.h>
int main (void)
{
time_t t0;
t0 = time (NULL);
printf ("%s", asctime(localtime(&t0)));
return 0;
}
```

43. Program for fork

```
.
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
int main()
{
pid_t cid;
int a = 0;
if ( (cid=fork())>0 ) {
a++;
printf("Parent : a=%d\n",a);
printf("Parent : My Child PID = %d\n",cid);
printf("Parent : My PID = %d\n",getpid());
wait(NULL);
printf("Parent : child done\n");
puts("Parent : DONE");
}else{
printf("Child : a=%d\n",a);
printf("Child : My PID = %d\n",getpid());
printf("Child : My Parent PID = %d\n",getppid());
sleep(1);
}
return 0;
}
```

44. Process to get parent id

```
#include <stdio.h>
#include <unistd.h>
int main()
{
    pid_t pid = getpid();
    printf("My PID = %d\n",pid);
    printf("My Parent PID = %d\n",getppid());
    return 0;
}
```

45. Program to demonstrate pipe

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(void)
{
    pid_t childpid;
    int fd[2];
    char buf[10] = "Bye";
    //fd[0] is for reading
    //fd[1] is for writing
    if (pipe(fd) < 0){
        return 1;
    }
    childpid=fork();
    if (childpid > 0){
        //parent
        write(fd[1],"Hello",sizeof("Hello"));
    }else if (childpid == 0){
        //child
        read(fd[0],buf,sizeof(buf));
    }
    printf("CHILD PID = %d , buf = %s \n", childpid, buf);
    return 0;
}
```

46. Program for sigaction

```

#include <stdio.h>
#include <unistd.h>
#include <signal.h>
void sig_usr(int signo){
if(signo==SIGINT)
printf("Got SIGINT\n");
else
printf("Unknown signal number\n");
}
int main(void)
{
struct sigaction sa;
sa.sa_handler = sig_usr;
sigemptyset(&sa.sa_mask);
sa.sa_flags = 0;
if(sigaction(SIGINT, &sa, NULL) == -1) {
printf("Error creating SIG_INT\n");
return 1;
}
pause();
return 0;
}

```

47. Program to demonstrate signal

```

#include <stdio.h>
#include <unistd.h>
#include <signal.h>
void sig_usr(int signo){
if(signo==SIGINT)
printf("Got SIGINT\n");
else
printf("Unknown signal number\n");
}
int main(void)
{
if( signal(SIGINT, sig_usr) == SIG_ERR ) {
printf("Error creating SIG_INT\n");
return 1;
}
pause();
return 0;
}

```

48. Program to demonstrate sleep

```
#include <stdio.h>
#include <time.h>
#include <unistd.h>
int main (void)
{
    time_t t0;
    t0 = time(NULL);
    /* Better than loop */
    sleep(2);
    printf("Executed in %lf seconds.\n", difftime(time(NULL), t0));
    return 0;
}
```

49. Program to demonstrate watch function

```
#include <stdio.h>
#include <time.h>
int main (void)
{
    time_t t0;
    t0 = time (NULL);
    printf("%lu sec since epoch\n", t0);
    printf("%s\n", ctime(&t0));
    return 0;
}
```

50. Program for atoi

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char s[12] = "";
    int i = 0;
    scanf("%s",s);
    i = atoi(s);
    printf("%s = %i \n",s,i);
    return 0;
}
```

51. Program for fgets

```
.
#include <stdio.h>
int main()
{
    char a[8];
    fgets(a,sizeof(a),stdin);
    if( !a[sizeof(a)-1] ){
        printf("Greater than %lu. Thus, null terminated\n",sizeof(a));
    }
    puts(a);
    return 0;
}
```

52. Program for file write

```
#include <stdio.h>
int main()
{
    char* filename = "myfile.txt";
    FILE* fp = fopen(filename, "w");
    char c = 'A';
    if(fp){
        while( c<='Z' ){
            fputc(c,fp);
            c++;
        }
        fputc('\n',fp);
        fclose(fp);
    }else{
        printf("Opening %s failed\n", filename);
    }
    return 0;
}
```

53. Program to read file

```
#include <stdio.h>
int main()
{
    char* filename = "myFile.txt";
    FILE* fp = fopen(filename, "r");
    char c = 0;
    if(fp){
        while( (c=fgetc(fp)) != EOF ){
            printf("%c", c);
        }
    }
}
```

```

}
fclose(fp);
}else{
printf("Opening %s failed\n", filename);
}
puts("");
return 0;
}

```

54. Program for fscanf

```

#include <stdio.h>
int main()
{
char* filename = "numbers.txt";
FILE* fp = fopen(filename, "r");
int a = 0;
if(fp){
while( fscanf(fp,"%d",&a) != EOF ){
printf("%d\n", a);
}
fclose(fp);
}else{
printf("Opening %s failed\n", filename);
}
puts("");
return 0;
}

```

55. Program for fseek

```

.
#include <stdio.h>
int main()
{
char* filename = "myFile.txt";
FILE* fp = fopen(filename, "r");
char c = 0;
if(fp){
fseek(fp,5,SEEK_SET);
while( (c=fgetc(fp)) != EOF ){
printf("%c", c);
}
fclose(fp);
}else{
printf("Opening %s failed\n", filename);
}
}

```



```
puts("");
return 0;
}
```

56. Program for break

```
#include <stdio.h>
int main()
{
char a = 'A';
while(1){
printf("(Y)es, (N)o or (Q)uit\n");
printf("Enter choice: ");
scanf("%c",&a);
if(a=='q' || a=='Q'){
break;
}
}
return 0;
}
```

57. Program to demonstrate continue

```
#include <stdio.h>
int main()
{
int i = 0;
for(;i<100;i++){
if(i%2){
continue;
}
printf("%d*%d=%d\n",i,i,i*i);
}
return 0;
}
```

58. Program to show do_while

```
#include <stdio.h>
int main()
{
char a = 'A';
do{
printf("(Y)es, (N)o or (Q)uit\n");
printf("Enter choice: ");
scanf("%c",&a);
```

```

}while(a!='q' && a!='Q');
return 0;
}

```

59. Program for for loop

```

#include <stdio.h>
#include <math.h>
int main()
{
int a = 'A';
int b = 'a';
for(a='A',b='a'; a<='Z' && b<='z'; a++,b++ ){
printf("%c-%c\n",a,b);
}
return 0;
}

```

60. Program for pass by value and pass byreference

```

#include <stdio.h>
void inc1(int byval)
{
byval++;
}
void inc2(int *byref)
{
(*byref)++;
}
int main()
{
int a = 0;
int b = 0;
inc1(a);
inc2(&b);
printf("a= %d\n",a);
printf("b= %d\n",b);
return 0;
}

```

61. Program to show pointers in C

```

.
#include<stdio.h>
int main(){
float celsius, fahrenheit;
printf("Enter Fahrenheit: ");

```

```
scanf("%f", &fahrenheit);
celsius = (fahrenheit - 32) / 1.8;
printf("\nCelsius: %.2f\n", celsius);
return 0;
}
```

62. Program to print heart pattern

```
#include<stdio.h>
int main(){
int i,j;
//loop for the upper part of the heart
for(i=0;i<3;i++){
for(j=i;j<2;j++){
printf(" ");
}
for(j=(i*2)+5;j>0;j--){
printf("*");
}
for(j=5-(i*2);j>0;j--){
printf(" ");
}
for(j=(i*2)+5;j>0;j--){
printf("*");
}
printf("\n");
}
//loop for the lower part of the heart
for(i=0;i<10;i++){
for(j=0;j<i;j++){
printf(" ");
}
for(j=19-(2*i);j>0;j--){
printf("*");
}
printf("\n");
}
return 0;
}
```

63: Program for pyramid pattern

```
#include <stdio.h>
int main()
{
int i, space, rows, k=0, count = 0, count1 = 0;
printf("Enter number of rows: ");
```

```

scanf("%d",&rows);
for(i=1; i<=rows; ++i)
{
for(space=1; space <= rows-i; ++space)
{
printf(" ");
++count;
}
while(k != 2*i-1)
{
if (count <= rows-1)
{
printf("%d ", i+k);
++count;
}
else
{
++count1;
printf("%d ", (i+k-2*count1));
}
++k;
}
count1 = count = k = 0;
printf("\n");
}
return 0;
}

```

64: Program for Celsius to Fahrenheit

```

#include<stdio.h>
int main(){
float celsius, fahrenheit;
// Formula: T(°F) = T(°C) × 9/5 + 32 = Celsius * 1.8 + 32
printf("Enter Celsius: ");
scanf("%f", &celsius);
fahrenheit = celsius * 1.8 + 32;
printf("\nFahrenheit: %.2f\n", fahrenheit);
return 0;
}

```

65. Program to calculate equilateral triangle

```

#include<stdio.h>
#include<math.h>
int main(){
int side;

```

```
float area;
printf("Side: ");
scanf("%d", &side);
area = (sqrt(3) / 4)*(side*side);
printf("Area: %.2f", area);
printf("\n");
return 0;
}
```

66. Program to find area of dead body

```
#include<stdio.h>
int main(){
float length, breadth, area;
printf("Enter Length: ");
scanf("%f", &length);
printf("Enter Breadth: ");
scanf("%f", &breadth);
area = breadth * length;
printf("\nArea: %.2f\n", area);
return 0;
}
```

67. Area of rectangle

```
#include<stdio.h>
int main(){
int base, height, area;
printf("Base: ");
scanf("%d", &base);
printf("Height: ");
scanf("%d", &height);
area = (height * base) / 2;
printf("Area: %d", area);
printf("\n");
return 0;
}
```

68. Program to get day-month-year

```
#include<stdio.h>
int main(){
int year, month, week, days;
printf("Enter Days: ");
scanf("%d", &days);
year = (days - (days % 365)) / 365; // Get Year
days = (days % 365);
```

```

month = (days - (days % 30)) / 30; // Get Month
days = days % 30;
week = (days - (days % 7)) / 7; // Get Week
days = days % 7; // Days
printf("\n%d Year, %d Month, %d Week and %d Days\n", year, month, week, days);
return 0;
}

```

69 . Program to get diameter and circumference from radius

```

#include<stdio.h>
int main(){
float radius, diameter, circumference, area, pi;
pi = 3.1415926535;
printf("Enter Radius: ");
scanf("%f", &radius);
diameter = 2 * radius;
circumference = pi * diameter;
area = pi * (radius * radius);
printf("\nDiameter: %.8f", diameter);
printf("\nCircumference: %.8f", circumference);
printf("\nArea: %.8f", area);
printf("\n");
return 0;
}

```

70. Program for Fahrenheit to celsius

```

#include<stdio.h>
int main(){
float celsius, fahrenheit;
printf("Enter Fahrenheit: ");
scanf("%f", &fahrenheit);
celsius = (fahrenheit - 32) / 1.8;
printf("\nCelsius: %.2f\n", celsius);
return 0;
}

```

71. Program to check if file exist

```

#include <stdio.h>
int main()
{
FILE *file = fopen("Test.txt", "r");
if(file == NULL)
{
printf("ERROR : File Does Not Exist\n");
}
}

```

```

return -1;
}
else
{
printf("File Exists\n");
}
return 0;
}

```

72: Get file size

```

#include <stdio.h>
int main(int argc, char *argv[])
{
if(argc != 2)
{
printf("usage %s file.txt\n",argv[0]);
return -1;
}
FILE * file = fopen(argv[1], "r");
if(file == NULL)
{
printf("[ERROR] Unable to open %s\n",argv[1]);
return -1;
}
fseek(file, 0, SEEK_END);
printf("%s is %d bytes long\n",argv[1],ftell(file));
return 0;
}

```

73. Program for dynamic memory allocation

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main() {
char name[100];
char *description;
strcpy(name, "Imam Sutono");
// allocate memory dynamically
description = malloc(200 * sizeof(char));
if (description == NULL) {
fprintf(stderr, "Error - unable to allocate required memory");
} else {
strcpy(description, "Imam Sutono is student at Binus University");
}
printf("Name : %s\n", name);
}

```

```
printf("Description : %s\n", description);
return 0;
}
```

74. Program for resizing and releasing of memory

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main() {
char name[100];
char *description;
strcpy(name, "Imam Sutono");
// allocate memory dynamically
description = malloc(30 * sizeof(char));
if (description == NULL) {
fprintf(stderr, "Error - unable to allocate required memory");
} else {
strcpy(description, "He is student at Binus University");
}
printf("Name : %s\n", name);
printf("Description : %s\n", description);
// release memory using free() function
free(description);
}
```

75. Program for static variable

```
#include <stdio.h>
// function declaration
void func(void);
static int count = 5; // global variable
int main()
{
while(count--)
{
func();
}
return 0;
}
// function definition
void func(void)
{
static int i = 5; // local static variable
i++;
printf("i is %d and count is %d\n", i, count);
}
```


76. Program for case operator

```
#include <stdio.h>
int main() {
int sum = 17, count = 5;
double mean;
mean = (double) sum / count; // convert data type of sum (int) to double
printf("Value of sum : %f\n", mean);
return 0;
}
```

77. Program for integer promotion

```
#include <stdio.h>
int main() {
int i = 17;
char c = 'c'; // ASCII value is 99
int sum;
sum = i + c; // convert 'c' to ASCII value then performing addition operation
printf("Value of sum: %d\n", sum);
return 0;
}
```

78. Program for typedef

```
#include <stdio.h>
#include <string.h>
typedef struct Books {
char title[50];
char author[50];
char subject[100];
int book_id;
} Book;
int main() {
Book book;
strcpy(book.title, "C Programming");
strcpy(book.author, "ajay");
strcpy(book.subject, "C Programming Tutorial");
book.book_id = 6495407;
printf("Book 1 title : %s\n", book.title);
printf("Book 1 author : %s\n", book.author);
printf("Book 1 subject : %s\n", book.subject);
printf("Book 1 book_id : %d\n", book.book_id);
return 0;
}
```

79. Program for bitwise left shift operator

```
#include <stdio.h>
int main()
{
    int a=7, b=2,c;
    c = a<<b;
    printf("Value of c = %d\n",c);
    return 0;
}
```

80. Program for float

```
#include <stdio.h>
int main(void)
{
    const double RENT = 3852.99; // const-style constant
    printf("%f\n", RENT);
    printf("%e\n", RENT);
    printf("%.2f\n", RENT);
    printf("%.3f\n", RENT);
    printf("%.3f\n", RENT);
    printf("%.3E\n", RENT);
    printf("%.2f\n", RENT);
    printf("%.2f\n", RENT);
    return 0;
}
```

81. Program for bitwise complement

```
#include <stdio.h>
int main()
{
    int a=14, b;
    b = ~a;
    printf("Value of c = %d\n",b);
    return 0;
}
```

82. Program for bitwise and

```
.
#include <stdio.h>
int main()
{
    int a=14, b= 7, c;
    c =a&b;
```

```
printf("Value of c = %d\n",c);
return 0;
}
```

83. Program for getting remainder

```
#include<stdio.h>
int main ()
{
int t, A, B;
int rem = 0;
scanf ("%d", &t);
while (t--)
{
scanf ("%d%d",&A,&B);
rem = A % B;
printf("%d\n", rem);
}
return 0;
}
```

84. Program for ternary operator

```
#include<stdio.h>
int main()
{
float a,b,c,large;
printf("Enter any 3 numbers\n");
scanf("%f%f%f",&a,&b,&c);
large = a>b? (a>c?a:c): (b>c?b:c);
printf("The larger no is :%f\n", large);
return 0;
}
```

85. Program for structure in c

```
.
#include <stdio.h>
/*structure declaration*/
struct employee{
char name[30];
int empId;
float salary;
};
int main()
{
/*declare and initialization of structure variable*/
```

```

struct employee emp={"Mike",1120,76909.00f};
printf("\n Name: %s" ,emp.name);
printf("\n Id: %d" ,emp.empld);
printf("\n Salary: %f\n",emp.salary);
return 0;
}

```

86. Program for nested structure

```

#include <stdio.h>
struct student{
char name[30];
int rollNo;
struct dateOfBirth{
int dd;
int mm;
int yy;
}DOB; /*created structure varoable DOB*/
};
int main()
{
struct student std;
printf("Enter name: "); gets(std.name);
printf("Enter roll number: "); scanf("%d",&std.rollNo);
printf("Enter Date of Birth [DD MM YY] format: ");
scanf("%d%d%d",&std.DOB.dd,&std.DOB.mm,&std.DOB.yy);
printf("\nName : %s \nRollNo : %d \nDate of birth :
%02d/%02d/%02d\n",std.name,std.rollNo,std.DOB.dd,std.DOB.mm,std.DOB.yy);
return 0;
}

```

87. Program for structure pointer

```

#include <stdio.h>
struct item
{
char itemName[30];
int qty;
float price;
float amount;
};
int main()
{
struct item itm; /*declare variable of structure item*/
struct item *pltem; /*declare pointer of structure item*/
pltem = &itm; /*pointer assignment - assigning address of itm to pltem*/
/*read values using pointer*/
}

```

```

printf("Enter product name: ");
gets(pltem->itemName);
printf("Enter price:");
scanf("%f",&pltem->price);
printf("Enter quantity: ");
scanf("%d",&pltem->qty);
/*calculate total amount of all quantity*/
pltem->amount =(float)pltem->qty * pltem->price;
/*print item details*/
printf("\nName: %s",pltem->itemName);
printf("\nPrice: %f",pltem->price);
printf("\nQuantity: %d",pltem->qty);
printf("\nTotal Amount: %f",pltem->amount);
return 0;
}

```

88. Program for passing structure to function

```

.
#include<stdio.h>
struct address
{
char city[20];
int pin;
char phone[14];
};
struct employee
{
char name[20];
struct address add;
};
void display(struct employee);
void main ()
{
struct employee emp;
printf("Enter employee information?\n");
scanf("%s %s %d %s",emp.name,emp.add.city, &emp.add.pin, emp.add.phone);
display(emp);
}
void display(struct employee emp)
{
printf("Printing the details...\n");
printf("%s %s %d %s",emp.name,emp.add.city,emp.add.pin,emp.add.phone);
}

```

89. Program for union

```
#include <stdio.h>
union test1
{
int x;
int y;
} Test1;
union test2
{
int x;
char y;
} Test2;
union test3
{
int arr[10];
char y;
} Test3;
int main()
{
printf("sizeof(test1) = %lu, sizeof(test2) = %lu, "
"sizeof(test3) = %lu",
sizeof(Test1),
sizeof(Test2), sizeof(Test3));
return 0;
}
```

90. Program for union pointer

```
#include <stdio.h>
union test
{
int x;
char y;
};
int main()
{
union test p1;
p1.x = 65;
// p2 is a pointer to union p1
union test* p2 = &p1;
// Accessing union members using pointer
printf("%d %c", p2->x, p2->y);
return 0;
}
```

91. Program for bitfields

```
#include <stdio.h>
#include <string.h>
/* define simple structure */
struct {
    unsigned int widthValidated;
    unsigned int heightValidated;
} status1;
/* define a structure with bit fields */
struct {
    unsigned int widthValidated : 1;
    unsigned int heightValidated : 1;
} status2;
int main( ) {
    printf( "Memory size occupied by status1 : %d\n", sizeof(status1));
    printf( "Memory size occupied by status2 : %d\n", sizeof(status2));
    return 0;
}
```