

## Basic

### Temperature from Celsius to Fahrenheit.

```
#include <stdio.h>
int main()
{
    int Celcious, Faren;
    float f( " .f ", & Celcious );
    scanf( " %f ", & Celcious );
    Faren = ( Celcious * 9 / 5 ) + 32;
    printf( " %.2f °C = %.2f °F in, %c Celcious, %c Faren );"
    return 0;
}
```

### Area and Perimeter calculator:

```
#include <stdio.h>
#include <math.h>
int main()
{
    int choice;
    float length, width, radius, base, height, i, n, j;
    float area, perimeter;
    printf( " AREA : & PERIMETER, " );
    printf( " 1. rectangle " );
    printf( " 2. circle " );
    printf( " 3. triangle " );
    printf( " Enter your choice: " );
    scanf( " %d ", & choice );
```

switch (choice)

Case 1:

```
printf("Enter length & width: ");
scanf("%f %f", &length, &width);
area = length * width;
Perimeter = 2 * (length + width);
printf("Rectangle Area: %.2f\n", area);
printf("Rectangle Perimeter: %.2f\n", Perimeter);
break;
```

Case 2:

```
printf("Enter radius: ");
scanf("%f", &radius);
area = 3.1416 * radius * radius;
Perimeter = 2 * 3.1416 * radius;
printf("Circle Area: %.2f\n", area);
printf("circle Perimeter: %.2f\n", Perimeter);
break;
```

Case 3:

```
printf("Enter base & height: ");
scanf("%f %f", &base, &height);
area = 0.5 * base * height;
```

```
Pointf("Enter three sides:");  
scanf("%f %f %f", &i, &j, &k);  
Perimeter = i + j + k;
```

```
printf("triangle Area: %.2f\n", area);  
printf("triangle Perimeter: %.2f\n", Perimeter);
```

```
break;
```

default:

```
printf("Invalid choice!\n");
```

```
return 0;
```

```
}
```

ASCII value of character:

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

```
}
```

```
:(
```

```
:(
```

II Goto statement:  
 II include <stdio.h>  
 int main()  
 int i = 1;  
 if (i%2 == 0){  
 Even!  
 printf("The Number is even\n");  
 goto END;  
 }  
 else {  
 ODD:  
 printf("The Number is odd\n");  
 }  
 END:  
 printf("End the Program");  
 return;  
}

int main() {

int a = 5;  
 int b =  $\frac{1+9*3}{2}$ ; if 18  
 int c = a  $\frac{6}{2}$  + b++;  
 }

## Operation:

- Minimum of two Number using Ternary Operator.

```
if included & did not include
int main()
{
    int a, b, min;
    scanf("%d %d", &a, &b);
    min = a < b ? a : b;
    printf("the minimum is %d", min);
    return 0;
}
```

Input: 4, 5  
Minimum: 4

- Find maximum of Three Number using

```
int main()
{
    int a, b, c, max;
    scanf("%d %d %d", &a, &b, &c);
    if ((a >= b) && (a >= c))
        max = a;
    else if ((b >= a) && (b >= c))
        max = b;
    else
        max = c;
}
```

```
printf("the Maximum of %d, %d, %d is: %d", a, b, c, max);
return 0;
```

- % 84

Even or odd, using modulo and bitwise operations:

```
#include <stdio.h>
int main()
{
    int num;
    scanf("%d", &num);
    if (num % 2 == 0)
        printf("Even modulo: %d in", num);
    else
        printf("Odd modulo: %d in", num);
    if (num & 1 == 0)
        printf("using bitwise Even: %d in", num);
    else
        printf("using bitwise Odd: %d in", num);
    return 0;
}
```

Swap two Numbers, without temporary variable:

```
#include <stdio.h>
int main()
{
    int a, b;
    scanf("%d %d", &a, &b);
    if (a > b)
        swap(a, b);
    printf("a = %d, b = %d", a, b);
}
```

a = a + b; "a/b" : forms

$$\frac{1}{b} = a - b; \quad \text{(D) } \text{not enough information}$$

$$\bar{a} = a - b; \quad \text{so} \quad \text{using } a = \sqrt{d}$$

```
a = a - b;  
printf ("After swapping: a=%d, b=%d\n", a, b);  
,
```

## Array swapping (Reverse Array) :

```
#include <stdio.h>
```

```
int main()
```

```
int n, i, sum = 0, temp;
```

```
scanf("%d", &n);
```

```
int arr[n];
```

```
printf("Enter %d number : ", n);
```

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

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

```
sum += arr[i];
```

```
for (i = 0; i < n/2; i++)
```

```
temp = arr[i];
```

```
arr[i] = arr[n - i - 1];
```

```
arr[n - i - 1] = temp;
```

```
printf("In Reversed Array: ");
```

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

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

```
printf("sum of : %d\n", sum);
```

```
return 0;
```

## Swapping - Two Number:

```
#include <stdio.h>
```

```
int main() {
```

```
    int a, b, temp;
```

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

```
    temp = a;
```

```
    a = b;
```

```
    b = temp;
```

```
    printf("After swapping: a=%d, b=%d\n", a, b);
```

```
    return 0;
```

## Condition

### Vowel, Consonant, digit, or special symbol:

```
#include <stdio.h>
```

```
#include <ctype.h>
```

```
int main() {
```

```
    char ch;
```

```
    scanf("%c", &ch);
```

```
    if (isalpha(ch)) {
```

```
        char lower = tolower(ch);
```

```
        char upper = toupper(ch);
```

```
        if ((lower == 'a' || lower == '@') || lower == 'e')
```

```
if (lower == 'O' || lower == 'U') && (upper == 'A' ||  
upper == 'E' || upper == 'I' || upper == 'O' ||  
upper == 'U') } q  
printf(" %c is vowel\n", ch); }  
else  
printf(" %c is consonant\n", ch); }  
else if (isdigit(ch)) {  
printf(" %c is digit\n", ch); }  
else  
printf(" %c is SPECIAL SYMBOL\n", ch);  
return 0;
```

if (ch & 0x20) {  
if ((ch & 0x0f) == 0x01) {  
if ((ch & 0x0f) == 0x05) {  
if ((ch & 0x0f) == 0x09) {  
if ((ch & 0x0f) == 0x0d) {

val ('A' == vowel || 'E' == vowel || 'I' == vowel || 'O' == vowel || 'U' == vowel) )) );

Calculation using switch case:

```
#include <stdio.h>
```

```
int main()
```

char opar;

```
double nu1, nu2, namur;
```

```
scanf("%c", &namur);
```

```
scanf("%f %f", &nu1, &nu2);
```

```
switch(namur){
```

Case '+':

```
namur = nu1 + nu2;
```

```
printf("%.2f + %.2f = %.2f\n", nu1, nu2, namur);
```

break;

Case '-':

```
namur = nu1 - nu2;
```

```
printf("%.2f - %.2f = %.2f\n", nu1, nu2, namur);
```

Case '\*':

```
namur = nu1 * nu2;
```

```
printf("%.2f * %.2f = %.2f\n", nu1, nu2, namur);
```

break;

Case '/':

```
namur = nu1 / nu2;
```

```
printf("%.2f / %.2f = %.2f\n", nu1, nu2, namur);
```

```

} else
    printf ("Error: It not allow '\n'"); // if
} break;
default:
    printf ("! Invalid Operator. In "); // mode
}
return 0;
}

```

## Greatest Three Number Using If else:

```

#include <stdio.h>
int main()
{
    int a, b, c;
    scanf ("%d %d %d", &a, &b, &c);
    if (a >= b && a >= c)
        printf ("%d is Greater\n", a);
    else if (b >= a && b >= c)
        printf ("%d is Greater\n", b);
    else
        printf ("%d is Greater\n", c);
}

```

```
printf("%d is Greater \n", c);  
return 0;  
}
```

| input: 3, 4  
| 5 is Greater

Goto for Repeated input until 0:

```
#include <stdio.h>
```

```
int main(){  
    int number, sum=0, count=0;
```

inputloop:

```
    printf("Enter a number: ");  
    scanf("%d", &number);
```

```
    if(number != 0){  
        sum += number;
```

Count ++;

goto inputloop; }

input:  
Enter a nube: 1  
" " : 2  
" " : 3  
" " : 0

output:

Total Number Enter: 3  
sum of all: 6  
Avarage: 2.00

```
    printf(" total number enter: %d \n", count);
```

```
    printf(" sum of Number: %d \n", sum);
```

```
    if(count > 0){
```

printf("Avarage: %.2f \n", (float) sum/count);

} else

```
        printf(" no number enter \n"); }
```

write a program to calculate & arrange 4  
 positive integers number. + input. 2 5 1 3 output  
 num: 10.0 avg = 10.0  
 num: -5 -  
 num: 20.0  $\left\{ \frac{(10+5+20)}{3} \right\}$  Avg = 15.0  
 -  
 include <stdio.h>  
 int main(){  
 float sum=0.0, num;  
 for(i=1; i<=6; i++){  
 printf(" num: ");  
 scanf("%f", &num);  
 if (num>0){  
 count++;  
 sum+=num;  
 }  
 if (count>1){  
 float avg = sum/count;  
 float (%.1f \n", Avg);  
 }  
 else {  
 printf("-\n");  
 }  
 }  
 return 0;

- C Programme to find sum of odd Digits.

```
#include <stdio.h>
int main()
{
    int number, digit, oddSum=0, evenSum=0;
    scanf("%d", &number);
    if (number<0)
        number = -number;
    while (number>0)
    {
        digit = number%10;
        if (digit % 2 == 0)
            evenSum += digit;
        else
            oddSum += digit;
        number /= 10;
    }
}
```

Input : 123456  
evenSum: 12  
oddSum: 9 .

- C Programme find the smaller number & largest number:

```
#include <stdio.h>
int main()
{
    int num, digit;
    int smaller=9, large=0;
    scanf("%d", &num);
    if (num<0) num = -num;
```

Find the Greatest Number and smallest number of 3 numbers

```
#include <stdio.h>
int main()
{
    int small, largest;
    int num1, num2, num3;
    scanf("%d %d %d", &num1, &num2, &num3);
    largest = num1;
    if (num2 > largest)
        largest = num2;
    if (num3 > largest)
        largest = num3;
    small = num1;
    if (num2 < small)
        small = num2;
    if (num3 < small)
        small = num3;
    printf("The greatest no: %d\n", largest);
    printf("The smallest no: %d", small);
}
```

input:  
1  
2  
3  
smaller: 2  
greater: 3

## Part 1:

### • Pyramid:

#include <stdio.h>

int main()

int row, space;

scanf ("%d", &row);

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

    for (space = 1; space <= row - i; space++) {

        printf (" ");

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

        printf ("\*");

    printf ("\n");

}

### • Diamond:

for (int i = row - 1; i >= 1; i--) {

    for (space = 1; space <= row - i; space++) {

        printf (" ");

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

        printf ("\*");

    printf ("\n"); }

### square :

```
#include <stdio.h>
```

```
int main()
```

```
int row;
```

```
scanf ("%d", &row);
```

```
for (int i=1; i<=row; i++) {
```

```
    for (int j=1; j<=row; j++) {
```

```
        if (i == 1 || i == row || j == 1 || j == row) {
```

```
            printf ("* ");
```

```
        else,
```

```
            printf (" ");
```

```
        printf ("\n");
```

### Pascal's Triangle:

```
#include <stdio.h>
```

```
int main()
```

```
int row, cofn=1;
```

```
scanf ("%d", &row);
```

```
for (int i=0; i<row; i++) {
```

```
    for (int space=0; space<=row-i; space++) {
```

```
        printf (" ");
```

```
    for (int j=0; j<=i; j++) {
```

input: 5  
output:

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

```
if(j==0 || i==0){
```

```
    cofn = 1;
```

```
}
```

```
else
```

```
    cofn = cofn * (i-j+1)/j;
```

```
printf("%d", cofn);
```

```
printf("\n");
```

```
}
```

input: 5  
output:

1 2 1

1 3 3 1  
1 4 6 4 1

### Floyd's Triangle:

```
#include <stdio.h>
```

```
int main()
```

```
int row, number=1;
```

```
scanf("%d", &row);
```

```
for(int i=1; i<=row; i++)
```

```
    for(int j=1; j<=i; j++)
```

```
        printf("%d", number);
```

```
        number++;
```

```
    printf("\n");
```

```
return 0;
```

input: 4  
output:

1  
2 3

4 5 6

7 8 9 10

### Right angle-triangle:

```
#include <stdio.h>
int main()
int row;
scanf("%d", &row);
for (int i = 0; i <= row; i++)
    for (int j = 0; j <= i; j++)
        printf("*");
    printf("\n");
return 0;
```

input: 5

output:

from for

(row + Long) \* (A - 1)

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

for (i > i < i) no

for (j > j < j) no

### inverse right triangle:

```
for (int i = row; i >= 1; i--) {
```

Pattan:

- Pattern 1s and 0s Using Parity check:

```
#include <stdio.h>
int main()
{
    int rows, i, j;
    scanf("%d", &rows);
    for(i=1; i<=rows; i++)
    {
        for(j=1; j<=i; j++)
        {
            if((i+j)%2 == 0)
                printf("1");
            else
                printf("0");
        }
    }
}
```

Input: 5  
Output: 10

~~17~~ ~~Friday~~ Jan 21

1 0 ) 1 0 1

0 1 0 1 0 1

10100101

22.  $\left( \frac{1}{2}x^2 + 3x - 5 \right) \cdot 60$

printf("%d");

1. S.Y.C. +

(+) - trinol.

— 1 —

input: 5

Output:

+ + ) q L

+ ) y 1 2

600 atoms + 213

1, p. 2001-2012

$(bx)$  ~~first~~

is a white

1920-1921  
1921-1922  
1922-1923  
1923-1924  
1924-1925  
1925-1926  
1926-1927  
1927-1928  
1928-1929  
1929-1930  
1930-1931  
1931-1932  
1932-1933  
1933-1934  
1934-1935  
1935-1936  
1936-1937  
1937-1938  
1938-1939  
1939-1940  
1940-1941  
1941-1942  
1942-1943  
1943-1944  
1944-1945  
1945-1946  
1946-1947  
1947-1948  
1948-1949  
1949-1950  
1950-1951  
1951-1952  
1952-1953  
1953-1954  
1954-1955  
1955-1956  
1956-1957  
1957-1958  
1958-1959  
1959-1960  
1960-1961  
1961-1962  
1962-1963  
1963-1964  
1964-1965  
1965-1966  
1966-1967  
1967-1968  
1968-1969  
1969-1970  
1970-1971  
1971-1972  
1972-1973  
1973-1974  
1974-1975  
1975-1976  
1976-1977  
1977-1978  
1978-1979  
1979-1980  
1980-1981  
1981-1982  
1982-1983  
1983-1984  
1984-1985  
1985-1986  
1986-1987  
1987-1988  
1988-1989  
1989-1990  
1990-1991  
1991-1992  
1992-1993  
1993-1994  
1994-1995  
1995-1996  
1996-1997  
1997-1998  
1998-1999  
1999-2000  
2000-2001  
2001-2002  
2002-2003  
2003-2004  
2004-2005  
2005-2006  
2006-2007  
2007-2008  
2008-2009  
2009-2010  
2010-2011  
2011-2012  
2012-2013  
2013-2014  
2014-2015  
2015-2016  
2016-2017  
2017-2018  
2018-2019  
2019-2020  
2020-2021  
2021-2022  
2022-2023  
2023-2024  
2024-2025  
2025-2026  
2026-2027  
2027-2028  
2028-2029  
2029-2030  
2030-2031  
2031-2032  
2032-2033  
2033-2034  
2034-2035  
2035-2036  
2036-2037  
2037-2038  
2038-2039  
2039-2040  
2040-2041  
2041-2042  
2042-2043  
2043-2044  
2044-2045  
2045-2046  
2046-2047  
2047-2048  
2048-2049  
2049-2050  
2050-2051  
2051-2052  
2052-2053  
2053-2054  
2054-2055  
2055-2056  
2056-2057  
2057-2058  
2058-2059  
2059-2060  
2060-2061  
2061-2062  
2062-2063  
2063-2064  
2064-2065  
2065-2066  
2066-2067  
2067-2068  
2068-2069  
2069-2070  
2070-2071  
2071-2072  
2072-2073  
2073-2074  
2074-2075  
2075-2076  
2076-2077  
2077-2078  
2078-2079  
2079-2080  
2080-2081  
2081-2082  
2082-2083  
2083-2084  
2084-2085  
2085-2086  
2086-2087  
2087-2088  
2088-2089  
2089-2090  
2090-2091  
2091-2092  
2092-2093  
2093-2094  
2094-2095  
2095-2096  
2096-2097  
2097-2098  
2098-2099  
2099-20100  
20100-20101  
20101-20102  
20102-20103  
20103-20104  
20104-20105  
20105-20106  
20106-20107  
20107-20108  
20108-20109  
20109-20110  
20110-20111  
20111-20112  
20112-20113  
20113-20114  
20114-20115  
20115-20116  
20116-20117  
20117-20118  
20118-20119  
20119-20120  
20120-20121  
20121-20122  
20122-20123  
20123-20124  
20124-20125  
20125-20126  
20126-20127  
20127-20128  
20128-20129  
20129-20130  
20130-20131  
20131-20132  
20132-20133  
20133-20134  
20134-20135  
20135-20136  
20136-20137  
20137-20138  
20138-20139  
20139-20140  
20140-20141  
20141-20142  
20142-20143  
20143-20144  
20144-20145  
20145-20146  
20146-20147  
20147-20148  
20148-20149  
20149-20150  
20150-20151  
20151-20152  
20152-20153  
20153-20154  
20154-20155  
20155-20156  
20156-20157  
20157-20158  
20158-20159  
20159-20160  
20160-20161  
20161-20162  
20162-20163  
20163-20164  
20164-20165  
20165-20166  
20166-20167  
20167-20168  
20168-20169  
20169-20170  
20170-20171  
20171-20172  
20172-20173  
20173-20174  
20174-20175  
20175-20176  
20176-20177  
20177-20178  
20178-20179  
20179-20180  
20180-20181  
20181-20182  
20182-20183  
20183-20184  
20184-20185  
20185-20186  
20186-20187  
20187-20188  
20188-20189  
20189-20190  
20190-20191  
20191-20192  
20192-20193  
20193-20194  
20194-20195  
20195-20196  
20196-20197  
20197-20198  
20198-20199  
20199-20200  
20200-20201  
20201-20202  
20202-20203  
20203-20204  
20204-20205  
20205-20206  
20206-20207  
20207-20208  
20208-20209  
20209-20210  
20210-20211  
20211-20212  
20212-20213  
20213-20214  
20214-20215  
20215-20216  
20216-20217  
20217-20218  
20218-20219  
20219-20220  
20220-20221  
20221-20222  
20222-20223  
20223-20224  
20224-20225  
20225-20226  
20226-20227  
20227-20228  
20228-20229  
20229-20230  
20230-20231  
20231-20232  
20232-20233  
20233-20234  
20234-20235  
20235-20236  
20236-20237  
20237-20238  
20238-20239  
20239-20240  
20240-20241  
20241-20242  
20242-20243  
20243-20244  
20244-20245  
20245-20246  
20246-20247  
20247-20248  
20248-20249  
20249-20250  
20250-20251  
20251-20252  
20252-20253  
20253-20254  
20254-20255  
20255-20256  
20256-20257  
20257-20258  
20258-20259  
20259-20260  
20260-20261  
20261-20262  
20262-20263  
20263-20264  
20264-20265  
20265-20266  
20266-20267  
20267-20268  
20268-20269  
20269-20270  
20270-20271  
20271-20272  
20272-20273  
20273-20274  
20274-20275  
20275-20276  
20276-20277  
20277-20278  
20278-20279  
20279-20280  
20280-20281  
20281-20282  
20282-20283  
20283-20284  
20284-20285  
20285-20286  
20286-20287  
20287-20288  
20288-20289  
20289-20290  
20290-20291  
20291-20292  
20292-20293  
20293-20294  
20294-20295  
20295-20296  
20296-20297  
20297-20298  
20298-20299  
20299-202100  
202100-202101  
202101-202102  
202102-202103  
202103-202104  
202104-202105  
202105-202106  
202106-202107  
202107-202108  
202108-202109  
202109-202110  
202110-202111  
202111-202112  
202112-202113  
202113-202114  
202114-202115  
202115-202116  
202116-202117  
202117-202118  
202118-202119  
202119-202120  
202120-202121  
202121-202122  
202122-202123  
202123-202124  
202124-202125  
202125-202126  
202126-202127  
202127-202128  
202128-202129  
202129-202130  
202130-202131  
202131-202132  
202132-202133  
202133-202134  
202134-202135  
202135-202136  
202136-202137  
202137-202138  
202138-202139  
202139-202140  
202140-202141  
202141-202142  
202142-202143  
202143-202144  
202144-202145  
202145-202146  
202146-202147  
202147-202148  
202148-202149  
202149-202150  
202150-202151  
202151-202152  
202152-202153  
202153-202154  
202154-202155  
202155-202156  
202156-202157  
202157-202158  
202158-202159  
202159-202160  
202160-202161  
202161-202162  
202162-202163  
202163-202164  
202164-202165  
202165-202166  
202166-202167  
202167-202168  
202168-202169  
202169-202170  
202170-202171  
202171-202172  
202172-202173  
202173-202174  
202174-202175  
202175-202176  
202176-202177  
202177-202178  
202178-202179  
202179-202180  
202180-202181  
202181-202182  
202182-202183  
202183-202184  
202184-202185  
202185-202186  
202186-202187  
202187-202188  
202188-202189  
202189-202190  
202190-202191  
202191-202192  
202192-202193  
202193-202194  
202194-202195  
202195-202196  
202196-202197  
202197-202198  
202198-202199  
202199-202200  
202200-202201  
202201-202202  
202202-202203  
202203-202204  
202204-202205  
202205-202206  
202206-202207  
202207-202208  
202208-202209  
202209-202210  
202210-202211  
202211-202212  
202212-202213  
202213-202214  
202214-202215  
202215-202216  
202216-202217  
202217-202218  
202218-202219  
202219-202220  
202220-202221  
202221-202222  
202222-202223  
202223-202224  
202224-202225  
202225-202226  
202226-202227  
202227-202228  
202228-202229  
202229-202230  
202230-202231  
202231-202232  
202232-202233  
202233-202234  
202234-202235  
202235-202236  
202236-202237  
202237-202238  
202238-202239  
202239-202240  
202240-202241  
202241-202242  
202242-202243  
202243-202244  
202244-202245  
202245-202246  
202246-202247  
202247-202248  
202248-202249  
202249-202250  
202250-202251  
202251-202252  
202252-202253  
202253-202254  
202254-202255  
202255-202256  
202256-202257  
202257-202258  
202258-202259  
202259-202260  
202260-202261  
202261-202262  
202262-202263  
202263-202264  
202264-202265  
202265-202266  
202266-202267  
202267-202268  
202268-202269  
202269-202270  
202270-202271  
202271-202272  
202272-202273  
202273-202274  
202274-202275  
202275-202276  
202276-202277  
202277-202278  
202278-202279  
202279-202280  
202280-202281  
202281-202282  
202282-202283  
202283-202284  
202284-202285  
202285-202286  
202286-202287  
202287-202288  
202288-202289  
202289-202290  
202290-202291  
202291-202292  
202292-202293  
202293-202294  
202294-202295  
202295-202296  
202296-202297  
202297-202298  
202298-202299  
202299-202300  
202300-202301  
202301-202302  
202302-202303  
202303-202304  
202304-202305  
202305-202306  
202306-202307  
202307-202308  
202308-202309  
202309-202310  
202310-202311  
202311-202312  
202312-202313  
202313-202314  
202314-202315  
202315-202316  
202316-202317  
202317-202318  
202318-202319  
202319-202320  
202320-202321  
202321-202322  
202322-202323  
202323-202324  
202324-202325  
202325-202326  
202326-202327  
202327-202328  
202328-202329  
202329-202330  
202330-202331  
202331-202332  
202332-202333  
202333-202334  
202334-202335  
202335-202336  
202336-202337  
202337-202338  
202338-202339  
202339-202340  
202340-202341  
202341-202342  
202342-202343  
202343-202344  
202344-202345  
202345-202346  
202346-202347  
202347-202348  
202348-202349  
202349-202350  
202350-202351  
202351-202352  
202352-202353  
202353-202354  
202354-202355  
202355-202356  
202356-202357  
202357-202358  
202358-202359  
202359-202360  
202360-202361  
202361-202362  
202362-202363  
202363-202364  
202364-202365  
202365-202366  
202366-202367  
202367-202368  
202368-202369  
202369-202370  
202370-202371  
202371-202372  
202372-202373  
202373-202374  
202374-202375  
202375-202376  
202376-202377  
202377-202378  
202378-202379  
202379-202380  
202380-202381  
202381-202382  
202382-202383  
202383-202384  
202384-202385  
202385-202386  
202386-202387  
202387-202388  
202388-202389  
202389-202390  
202390-202391  
202391-202392  
202392-202393  
202393-202394  
202394-202395  
202395-202396  
202396-202397  
202397-202398  
202398-202399  
202399-202400  
202400-202401  
202401-202402  
202402-202403  
202403-202404  
202404-202405  
202405-202406  
202406-202407  
202407-202408  
202408-202409  
202409-202410  
202410-202411  
202411-202412  
202412-202413  
202413-202414  
202414-202415  
202415-202416  
202416-202417  
202417-202418  
202418-202419  
202419-202420  
202420-202421  
202421-202422  
202422-202423  
202423-202424  
202424-202425  
202425-202426  
202426-202427  
202427-202428  
202428-202429  
202429-202430  
202430-202431  
202431-202432  
202432-202433  
202433-202434  
202434-202435  
202435-202436  
202436-202437  
202437-202438  
202438-202439  
202439-202440  
202440-202441  
202441-202442  
202442-202443  
202443-202444  
202444-202445  
202445-202446  
202446-202447  
202447-202448  
202448-202449  
202449-202450  
202450-202451  
202451-202452  
202452-202453  
202453-202454  
202454-202455  
202455-202456  
202456-202457  
202457-202458  
202458-202459  
202459-202460  
202460-202461  
202461-202462  
202462-202463  
202463-202464  
202464-202465  
202465-202466  
202466-202467  
202467-202468  
202468-202469  
202469-202470  
202470-202471  
202471-202472  
202472-202473  
202473-202474  
202474-202475  
202475-202476  
202476-202477  
202477-202478  
202478-202479  
202479-202480  
202480-202481  
202481-202482  
202482-202483  
202483-202484  
202484-202485  
202485-202486  
202486-202487  
202487-202488  
202488-202489  
202489-202490  
202490-202491  
202491-202492  
202492-202493  
202493-202494  
202494-202495  
202495-202496  
202496-202497  
202497-202498  
202498-202499  
202499-202500  
202500-202501  
202501-202502  
202502-202503  
202503-202504  
202504-202505  
202505-202506  
202506-202507  
202507-202508  
202508-202509  
202509-202510  
202510-202511  
202511-202512  
202512-202513  
202513-202514  
202514-202515  
202515-202516  
202516-202517  
202517-202518  
202518-202519  
202519-202520  
202520-202521  
202521-202522  
202522-202523  
202523-202524  
202524-202525  
202525-202526  
202526-202527  
202527-202528  
202528-202529  
202529-202530  
202530-202531  
202531-202532  
202532-202533  
202533-202534  
202534-202535  
202535-202536  
202536-202537  
202537-202538  
202538-202539  
202539-202540  
202540-202541  
202541-202542  
202542-202543  
202543-202544  
202544-202545  
202545-202546  
202546-202547  
202547-202548  
202548-202549  
202549-202550  
202550-202551  
202551-202552  
202552-202553  
202553-202554  
202554-202555  
202555-202556  
202556-202557  
202557-202558  
202558-202559  
202559-202560  
202560-202561  
202561-202562  
202562-202563  
202563-202564  
202564-202565  
202565-202566  
202566-202567  
202567-202568  
202568-202569  
202569-202570  
202570-202571  
202571-202572  
202572-202573  
202573-202574  
202574-202575  
202575-202576  
202576-202577  
202577-202578  
202578-202579  
202579-202580  
202580-202581  
202581-202582  
202582-202583  
202583-202584  
202584-202585  
202585-202586  
202586-202587  
202587-202588  
202588-202589  
202589-202590  
202590-202591  
202591-202592  
202592-202593  
202593-202594  
202594-202595  
202595-202596  
202596-202597  
202597-202598  
202598-202599  
202599-202600  
202600-202601  
202601-202602  
202602-202603  
202603-202604  
202604-202605  
202605-202606  
202606-202607  
202607-202608  
202608-202609  
202609-202610  
202610-202611  
202611-202612  
202612-202613  
202613-202614  
202614-202615  
202615-202616  
202616-202617  
202617-202618  
2026

= goal taxi

= 6166 + 111

## Number up to

1. *Leucosia* sp. (Diptera: Syrphidae) was collected from a small stream in the forest.

*(Continued from back cover)*

- The column ( $j$ ) is odd (1) or even (0)
    - #include <stdio.h>
- ```

int main()
{
    int rows, i, j;
    scanf("%d", &rows);
    for(i = 1; i <= rows; i++)
        for(j = 1; j <= i; j++)
            if(j % 2 != 0) printf("1");
            else printf("0");
    return 0;
}
  
```
- input: 3      output:  
                   1 0 1  
                   1 0 1 0 1  
                   1 0 1 0 1 0 1
- #include <stdio.h>
- ```

int main()
{
    int rows, i, j, start_Nu;
    scanf("%d", &rows);
    start_Nu = 2;
    for(i = 1; i <= rows; i++)
        for(j = 1; j <= i; j++)
            printf("%d", start_Nu);
    return 0;
}
  
```
- input: 3      output:  
                   2  
                   3 4  
                   5 6 7  
                   8 9 10 11

```
printf ("In");
```

```
    . . . ;
```

```
    }  
return 0;
```

```
}
```

## Loop

- Print Numbers 1 - 100 using all three loops.

```
#include <stdio.h>
int main()
{
    int i;
    for(i=1; i<=100; i++)
        printf("%d", i);
    while(i<=100)
        printf("%d", i);
    i=1;
    do
        printf("%d", i);
        i++;
    while(i<=100);
```

- Sum of First N nature Number:

```
int main()
{
    int n, sum=0;
    scanf("%d", &n);
    for(i=1; i<=n; i++)
        sum+=i;
    printf("sum of %d nature number = %d", n, sum);
    return 0;
}
```

### Factorial Number:

```

int main() {
    int i, n;
    long long fact = 1;
    scanf("%d", &n);
    if (n < 0) {
        printf("Negative Not Factorial");
    } else {
        for (i = 1; i <= n; i++) {
            fact *= i;
        }
        printf("Factorial is %d = %ld\n", n, fact);
    }
    return 0;
}

```

### Reverse Number:

```

#include <stdio.h>
int main () {
    int num, reverse = 0, remain;
    scanf("%d", &num);
    int original = num;
    while (num != 0) {
        remainder = num % 10;
        reverse = reverse * 10 + remainder;
        num = num / 10;
    }
    printf("Reverse number %d = %d\n", original, reverse);
}

```

Input: 125  
Output: 521

• Count Digits & Sum of Digit and Reverse  
 ii) include <stdio.h>  
 int main()

```

    int num, rem, sum_of_digit_and_reverse;
    scanf ("%d", &num);           Input: 1234
    int temp = num;              Output:
    while (temp > 0)
        rem = temp % 10;         Number of digits of
        sum += rem;              sum of a number
        reverse = reverse * 10 + rem; Reverse of a number
        temp = temp / 10;
        digit++;
    }
    printf ("Number of Digit : %d ; digit");
    printf (" sum of digit : %d ; sum");
    printf (" Reverse of Digit : %d ; reverse");
    }
  
```

### Multiplication Table:

```

    int main()
    {
        int num;
        Scanf ("%d", &num);
        for (i = 1; i <= 10; i++)
            printf ("%d x %d = %d\n", num, i, num * i);
    }
  
```

## Even numbers between limits and sum:

```
#include <stdio.h>
```

```
int main()
```

```
int start, end, i, sum = 0;
```

```
scanf("%d %d", &start, &end);
```

```
for (i = start; i <= end; i++)
```

```
    if (i % 2 == 0)
```

```
        printf("%d", i);
```

```
        sum += i;
```

```
printf(" even digit : %d \n", i);
```

```
}
```

```
printf("sum of even : %d \n", sum);
```

```
return 0;
```

## Prime Number:

```
#include <stdio.h>
```

```
int main()
```

```
int m, prime = 1;
```

```
scanf("%d", &m);
```

```
if (m < 2) {
```

```
    prime = 0;
```

else

if (m > 1 & prime == 1)

for (int i = 2; i <= m/2; i++)

if (m % i == 0)

{ prime = 0; break; }

if (prime == 1)

printf("%d is prime number", m);

else

## Even numbers between limits and sum:

```
#include <stdio.h>
```

```
int main()
```

```
int start, end, i, sum = 0;  
scanf("%d %d", &start, &end);
```

```
for (i = start; i <= end; i++) {
```

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

```
        printf("%d", i);  
        sum += i;
```

```
    } else {
```

```
        printf("sum of even : %d\n", sum);
```

```
    }
```

## Prime Number:

```
#include <stdio.h>
```

```
int main()
```

```
int m, prime = 1;
```

```
scanf("%d", &m);
```

```
if (m < 2) {
```

```
    prime = 0;
```

```
: else {
```

```
    for (int i = 2; i <= m/2; i++) {
```

```
        if (m % i == 0) {
```

```
            prime = 0;
```

```
        } else {
```

```
            prime = 1;
```

## Prime Numbers:

```
#include <stdio.h>
```

```
int main()
```

```
int num, i, count = 0;
```

```
scanf("%d", &num);
```

```
if (num <= 1)
```

```
return 0;
```

```
for (i = 1; i <= num; i++)
```

```
if (num % i == 0)
```

```
Count++;
```

```
if (Count == 2)
```

```
printf("%d is a Prime Number\n", num);
```

```
else
```

```
printf("%d is not prime\n", num);
```

```
}
```

6, 28, 496, 8128 perfect

## • Perfect Numbers:

```
#include <stdio.h>
```

```
int main()
```

```
int num, i, sum = 0;
```

```
scanf("%d", &num);
```

```
for (i = 1; i < num / 2; i++)
```

```
if (num % i == 0)
```

```
sum += i;
```

```
}
```

```
if (sum == num)
```

```
printf("%d is Perfect Number: %d", num, num);
```

```
else
```

```
printf("%d is Not Perfect Number: %d", num, num);
```

```
return 0;
```

11 21 11+4(+5)

1, 2, 145.

### Strong Numbers:

```
#include <stdio.h>
int main() {
    int num, temp, digit, fact, i, sum=0; // input: 2
    scanf ("%d", &num); // output: 2 is not
    temp = num; // Strong Number
    while (temp > 0) // get last digit
        digit = temp % 10; // if (sum == num)
        fact = 1; // if (sum == num)
        for (i=1; i <= digit; i++) fact *= i; // if (sum == num)
        sum += fact; // if (sum == num)
        temp = temp / 10; // if (sum == num)
    if (sum == num) printf ("%d is Strong Number");
    else printf ("%d is not Strong Number");
}
```

### Check Spy Number:

```
int main() {
    int num, digit, sum=0, product=1, temp;
    scanf ("%d", &num);
    temp = num;
    while (temp > 0) {
        digit = temp % 10;
        sum += digit;
        product *= digit;
        temp = temp / 10;
    }
    if (sum == product) // 123, 1124
        printf ("%d is Spy Number");
    else
        printf ("%d is not Spy Number");
}
```

```

if(sum == product)
    printf(" %d is Spy Number\n", num);
else
    printf("%d Is Not Spy Number\n", num);
return 0;
}

```

### • Palindrome Number:

```

int main()
{
    int num, revers=0, original, remain;
    scanf("%d", &num); input: 121
    original = num;
    while(num>0) { output: 121
        remain = num%10;
        revers = revers*10+remain;
        num = num/10;
        if(original == reversed) { Palindrome
            printf("%d is Palindrom ", original);
        }
        else
            printf("%d is Not Palindrom ", original);
    }
    return 0;
}

```

## Array:

- Number of Array sum:

```
#include <stdio.h>
int main()
int arr[100], sum=0;
for(i=0; i<5; i++)
    Scanf ("%d", &arr[i]);
    sum += arr[i];
```

input: 2, 3, 4, 5, 6,

sum: 20

```
Printf ("Sum of array: %d\n", sum);
```

```
return 0;
```

```
}
```

- Read and print one-dimensional array:

```
#include <stdio.h>
int main()
int arr[100], n, i;
Scanf ("%d", &n);
for(i=0; i<n; i++)
    Scanf ("%d", &arr[i]);
```

input: 3  
1, 2, 3

Output:

1, 2, 3

```
    Scanf ("%d", &arr[i]);
```

```
Printf ("Array element: ");
```

```
for(i=0; i<n; i++)
    Printf ("%d", arr[i]);
```

```
}
```

PATTERN.

- Find max and min:

```
#include <stdio.h>
int main()
int arr[100], n, i, maximum;
scanf("%d", &n);
for(i=0; i<n; i++) {
    scanf("%d", &arr[i]);
    if (arr[i] > maximum) maximum = arr[i];
    if (arr[i] < minimum) minimum = arr[i];
}
printf("maximum: %d\n", maximum);
printf("minimum: %d\n", minimum);
```

input: 3

1, 89 5

maximum: 89

minimum: 1

- reverse Array:

```
int main()
int arr[50], n, i, temp;
scanf("%d", &n);
for(i=0; i<n; i++) {
    scanf("%d", &arr[i]);
}
for(i=0; i<n/2; i++) {
    temp = arr[i];
    arr[i] = arr[n-i-1];
    arr[n-i-1] = temp;
}
```

arr[n-i-1] = temp;

}

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

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

}

return 0;

input: 3  
4, 5, 6  
output:  
6, 5, 4

## Function:

### Calculate Sum of two Number:

```
#include <stdio.h>
int addNumber(int a, int b) {
    int sum = a + b;
    return sum;
}
```

```
int main() {
    int num1, num2, result;
    scanf("%d %d", &num1, &num2);
    result = addNumber(num1, num2);
    printf("sum of: %d + %d = %d\n", num1, num2, result);
    return 0;
}
```

### Check Even or odd Number:

```
#include <stdio.h>
int checkEvenOdd(int number) {
    if (number % 2 == 0) {
        printf("%d is Even\n", number);
    } else {
        printf("%d is Odd", number);
    }
}
```

```
int main() {
    int num;
    scanf("%d", &num);
    checkEvenOdd(num);
}
```

input: 13, 25  
sum:  $13 + 25 = 38$

input: 5  
5 is odd

## Solution:

### • Compute Factorial:

```
#include <stdio.h>
```

```
int calculate(int n);
```

```
int fact = 1;
```

```
for (int i = 1; i <= n; i++) {  
    fact = fact * i;  
}
```

```
return fact;
```

```
int main()
```

```
int num;
```

```
scanf("%d", &num);
```

```
int result = calculate(num);
```

```
printf("Fractional %d / %d", num, result);
```

```
return 0;
```

Input: 5  
Output: 120

### • Check Prime Numbers:

```
#include <stdio.h>
```

```
int Prime(int num);
```

```
if (num <= 1) return 0;
```

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

```
    if (num % i == 0)
```

```
        return 0;
```

```
    return 1;
```

Input: 3

Output: 3

3 is a prime number

```
int main()
{
    int number;
    scanf ("%d", &number);
    if (Prime (number))
        printf ("%d is a prime number\n", number);
    else
        printf ("%d is not a prime number\n", number);
    return 0;
}
```

• Function Return Student Grade

```
#include <stdio.h>
int Calculate (int marks)
{
    if (marks >= 90)
        return 'A';
    else if (marks >= 80)
        return 'B';
    else if (marks >= 50)
        return 'C';
    else if (marks >= 40)
        return 'D';
    else
        return 'F';
}
```

```
int main()
{
    int marks;
    scanf ("%d", &marks);
    char grade = Calculate(marks);
    printf ("Marks: %d\n", marks);
    printf ("Grade: %.1f\n", grade);
    return 0;
}
```