



Patuakhali Science and Technology University.

Submitted By:

S M Arifuzzaman

ID :2302054

REG:11832

Submitted To:

Md. Mahbubur Rahman

Assistant Professor

Department of Computer Science and

Information Technology(CSIT)

Faculty of Computer Science and Engineering.

Date: 12 December, 2024

1. Print name, DOB, and mobile number

Write a C program to print your name, date of birth, and mobile number.

Code:

```
#include <stdio.h>

int main() {

    printf("Name\t: SM Arifuzzaman\n");

    printf("DOB\t: May 4, 2004\n");

    printf("Mobile\t: 01779182553\n");

    return 0;

}
```

Expected Output:

Name : SM Arifuzzaman
DOB : May 04, 2004
Mobile : 01779182553

2. Get the C version in use

Write a C program to get the C version you are using.

Code:

```
#include <stdio.h>

int main() {

    if(__STDC_VERSION__ >= 201710L) {

        printf("We are using C18!\n");

    }

}
```

```

else if(__STDC_VERSION__ >= 201111){

    printf("We are using C11!\n");

}

else if(__STDC_VERSION__ >= 199901L){

    printf("We are using C99!\n");

}

else if(__STDC_VERSION__ >= 201710L){

    printf("We are using C89/90!\n");

}


return 0;

}

```

Expected Output:

We are using C11!

3. Print block 'F' and a large 'C'

Write a C program to print a block F using the hash (#), where the F has a height of six characters and width of five and four characters. And also print a very large 'C'.

Code:

```

#include <stdio.h>


int main() {

    printf("#####\n#\n#\n#####\n#\n#\n#\n");
}

```

```

printf("    ##### \n");

printf("  ##      ##\n");

printf("##\n#\n#\n#\n#\n");

printf("  ##      ##\n");

printf("    ##### \n");

return 0;

}

```

Expected Output:

```

#####
#
#
#####
#
#
#

#####
##  ##
#
#
#
#
#
##  ##
#####

```

4. Reverse characters ('X', 'M', 'L')

Write a C program to print the following characters in reverse.

Test Characters: 'X', 'M', 'L'

Code:

```
#include <stdio.h>

int main()
{
    char ch1 = 'X';

    char ch2 = 'M';

    char ch3 = 'L';

    printf("The reverse of %c%c%c is %c%c%c\n",ch1,ch2,ch3,ch3, ch2,
ch1);

    return(0);
}
```

Expected Output:

The reverse of XML is LMX

5. Compute rectangle perimeter and area

Write a C program to compute the perimeter and area of a rectangle with a height of 7 inches and width of 5 inches.

Expected Output:

Perimeter of the rectangle = 24 inches

Area of the rectangle = 35 square inches

Code:

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

```
int main() {
```

```
int height = 7;

int width = 5;

int perimeter = 2 * (height + width);

int area = height * width;

printf("Perimeter of the rectangle = %d inches\n", perimeter);

printf("Area of rectangle = %d square inches\n", area);

return 0;

}
```

6. Compute circle perimeter and area

Write a C program to compute the perimeter and area of a circle with a given radius.

Code:

```
#include <stdio.h>

#include <math.h>

#define PI 3.14

int main() {

    float radius, perimeter, area;

    printf("Enter radius: ");

    scanf("%f", &radius);


    perimeter = 2 * PI * radius;

    area = PI * radius * radius;
```

```

printf("Perimeter of the circle: %f inches\n", perimeter);

printf("Area of the circle: %f square inches\n", area);

return 0;

}

```

Expected Output:

Perimeter of the Circle = 37.680000 inches

Area of the Circle = 113.040001 square inches

7. Display multiple variables of various types

Write a C program to display multiple variables.

Sample *Variables* :

a + c, x + c, dx + x, ((int) dx) + ax, a + x, s + b, ax + b, s + c, ax + c, ax + ux

Declaration :

int a = 125, b = 12345;

long ax = 1234567890;

short s = 4043;

float x = 2.13459;

double dx = 1.1415927;

char c = 'W';

unsigned long ux = 2541567890;

Code

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

```
int main()
```

```
{
```

```
    int a = 125, b = 12345;
```

```

long ax = 1234567890;

short s = 4043;

float x = 2.13459;

double dx = 1.1415927;

char c = 'W';

unsigned long ux = 2541567890;

printf("a + c = %d\n", a + c);

printf("x + c = %f\n", x + c);

printf("dx + x = %f\n", dx + x);

printf("((int) dx) + ax = %ld\n", ((int) dx) + ax);

printf("a + x = %f\n", a + x);

printf("s + b = %d\n", s + b);

printf("ax + b = %ld\n", ax + b);

printf("s + c = %hd\n", s + c);

printf("ax + c = %ld\n", ax + c);

printf("ax + ux = %lu\n", ax + ux);

return 0;

}

```

8. Convert days to years, weeks, and days

Write a C program to convert specified days into years, weeks and days.

Note: Ignore leap year.

Code:


```
#include <stdio.h>

int main()
{
    int days;

    printf("Number of days: ");

    scanf("%d", &days);

    int years = 0;

    int weeks = 0;

    while(days>=30) {

        if(days >= 365) {

            years = days / 365;

            days %= 365;

            continue;

        }

        if(days >= 7) {

            weeks = days / 7;

            days %= 7;

            continue;

        }

    }

    printf("Years: %d\nWeeks: %d\nDays: %d", years, weeks, days);

    return 0;
}
```

```
}
```

Test Data :

Number of days : 1329

Expected Output :

Years: 3

Weeks: 33

Days: 3

9. Calculate sum of two integers

Write a C program that accepts two integers from the user and calculates the sum of the two integers.

Code:

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

```
int main()
```

```
{
```

```
    int a, b;
```

```
    printf("Input the first integer: ");
```

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

```
    printf("Input the second integer: ");
```

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

```
    int sum = a + b;
```

```
    printf("Sum of the above two integers= %d\n", sum);
```

```
    return 0;
```

```
}
```

Test Data :

Input the first integer: 25

Input the second integer: 38

Expected Output:

Sum of the above two integers = 63

10. Calculate product of two integers

Write a C program that accepts two integers from the user and calculates the product of the two integers.

Code:

```
// Product of two integers

#include <stdio.h>

int main()
{
    int a, b;

    printf("Input the first integer: ");
    scanf("%d", &a);

    printf("Input the second integer: ");
    scanf("%d", &b);

    int product = a * b;

    printf("Product of the above two integers= %d\n", product);
```

```
    return 0;
}
```

Test Data :

Input the first integer: 25

Input the second integer: 15

Expected Output:

Product of the above two integers = 375

11. Calculate average weight for purchases

Write a C program that accepts two item's weight and number of purchases (floating point values) and calculates their average value.

Code:

```
#include <stdio.h>

int main() {

    double w1, c1, w2, c2, result;

    printf("Weight - Item1: ");

    scanf("%lf", &w1);

    printf("No. of item1: ");

    scanf("%lf", &c1);

    printf("Weight - Item2: ");

    scanf("%lf", &w2);

    printf("No. of item2: ");

    scanf("%lf", &c2);

    result = ((w1 * c1) + (w2 * c2)) / (c1 + c2);
```

```
    printf("Average Value = %f\n", result);  
  
    return 0;  
  
}
```

Test Data :

Weight - Item1: 15

No. of item1: 5

Weight - Item2: 25

No. of item2: 4

Expected Output:

Average Value = 19.444444

12. Print employee ID and monthly salary

Write a C program that accepts an employee's ID, total worked hours in a month and the amount he received per hour. Print the ID and salary (with two decimal places) of the employee for a particular month.

Code:

```
#include <stdio.h>  
  
int main()  
{  
  
    char ID[11];  
  
    int hours;  
  
    double amountPerHour, salary;  
  
    printf("Input the Employees ID(Max. 10 chars): ");
```

```

scanf("%10s", &ID);

printf("Input the working hrs: ");

scanf("%d", &hours);

printf("Salary amount/hr: ");

scanf("%lf", &amountPerHour);

salary = amountPerHour * hours;

printf("Employee ID: %s\n", ID);

printf("Salary = U$ %.2lf\n", salary);

return 0;
}

```

Test Data :

Input the Employees ID(Max. 10 chars): 0342

Input the working hrs: 8

Salary amount/hr: 15000

Expected Output:

Employees ID = 0342

Salary = U\$ 120000.00

13. Find the maximum of three integers

Write a C program that accepts three integers and finds the maximum of three.

Code:

```
#include <stdio.h>

int main()
{
    int a, b, c;

    printf("Enter the first integer: "); scanf("%d", &a);
    printf("Enter the second integer: "); scanf("%d", &b);
    printf("Enter the third integer: "); scanf("%d", &c);

    int max = a;

    if(b > max) max = b;

    if(c > max) max = c;

    printf("Maximum value of three integers: %d\n", max);

    return 0;
}
```

Test Data :

Input the first integer: 25

Input the second integer: 35

Input the third integer: 15

Expected Output:

Maximum value of three integers: 35

14. Calculate bike's average consumption

Write a C program to calculate a bike's average consumption from the given total distance (integer value) travelled (in km) and spent fuel (in liters, float number – 2 decimal points).

Code:

```
#include <stdio.h>

int main()
{
    double distance, fuel, avgConsume;

    printf("Input total distance in km: "); scanf("%lf", &distance);
    printf("Input total fuel spent in liters: "); scanf("%lf", &fuel);

    avgConsume = distance/fuel;

    printf("Average consumption (km/li): %.3lf", avgConsume);

    return 0;
}
```

Test Data :

Input total distance in km: 350

Input total fuel spent in liters: 5

Expected Output:

Average consumption (km/lt) 70.000

15. Calculate distance between two points

Write a C program to calculate the distance between two points.

Code:

```
// Calculate distance two points

#include <stdio.h>

#include <math.h>

int main()

{

    double x1, y1, x2, y2;

    double distance;

    printf("Input x1: ");scanf("%lf",&x1);

    printf("Input y1: ");scanf("%lf",&y1);

    printf("Input x2: ");scanf("%lf",&x2);

    printf("Input y2: ");scanf("%lf",&y2);

    distance = sqrt(pow((x2-x1), 2) + pow((y2-y1), 2));
```

```
    printf("Distance between the said points: %.4lf\n", distance);

    return 0;

}
```

Test Data :

Input x1: 25

Input y1: 15

Input x2: 35

Input y2: 10

Expected Output:

Distance between the said points: 11.1803

16. Break amount into smallest banknotes

Write a C program to read an amount (integer value) and break the amount into the smallest possible number of bank notes.

Code:

```
#include <stdio.h>

int main()
{
    int amount;

    int notes[] = {100, 50, 20, 10, 5, 2, 1};

    int count[7] = {0};
```

```

printf("Input the amount: ");

scanf("%d", &amount);

for(int i = 0; i < 7; i++){

    if(amount >= notes[i]){

        count[i] = amount/notes[i];

        amount = amount % notes[i];

    }

}

printf("There are: \n");

for(int i = 0; i < 7; i++){

    printf("%d Note(s) of %.2f\n", count[i], (float)notes[i]);

}

return 0;
}

```

Test Data :

Input the amount: 375

Expected Output:

There are:

3 Note(s) of 100.00

1 Note(s) of 50.00

1 Note(s) of 20.00

0 Note(s) of 10.00

1 Note(s) of 5.00

0 Note(s) of 2.00

0 Note(s) of 1.00

17. Convert seconds into hours, minutes, seconds

Write a C program to convert a given integer (in seconds) to hours, minutes and seconds.

Code:

```
#include <stdio.h>

int main() {

    int totalSeconds, hours, minutes, seconds;

    printf("Input seconds: "); scanf("%d", &totalSeconds);

    hours = totalSeconds / 3600;

    totalSeconds = totalSeconds % 3600;

    minutes = totalSeconds / 60;

    seconds = totalSeconds % 60;

    printf("There are:\nH:M:S - %d:%d:%d\n", hours, minutes, seconds);
```

```
    return 0;  
}
```

Test Data :

Input seconds: 25300

Expected Output:

There are:

H:M:S - 7:1:40

18. Convert days to years, months, days

Write a C program to convert a given integer (in days) to years, months and days, assuming that all months have 30 days and all years have 365 days.

Code:

```
#include <stdio.h>  
  
int main()  
{  
  
    int totalDays;  
  
    int years, months, days;  
  
    printf("Input no. of days: "); scanf("%d", &totalDays);  
  
    years = totalDays / 365;  
  
    totalDays %= 365;  
  
    months = totalDays / 30;
```

```
    days = totalDays % 30;

    printf("%d Year(s)\n", years);

    printf("%d Month(s)\n", months);

    printf("%d Day(s)\n", days);

    return 0;
}
```

Test Data :

Input no. of days: 2535

Expected Output:

6 Year(s)

11 Month(s)

15 Day(s)

19. Validate four integers based on conditions

Write a C program that accepts 4 integers p, q, r, s from the user where q, r and s are positive and p is even. If q is greater than r and s is greater than p and if the sum of r and s is greater than the sum of p and q print "Correct values", otherwise print "Wrong values".

Code:

```
#include <stdio.h>

int main()
```

```
{

    int p, q, r, s;

    printf("Input the first integer: ");

    scanf("%d", &p);

    printf("Input the second integer: ");

    scanf("%d", &q);

    printf("Input the third integer: ");

    scanf("%d", &r);

    printf("Input the fourth integer: ");

    scanf("%d", &s);


    if((q > r) && (s > p)

        && ((r+s) > (p+q))

        && (q>0)

        && (r>0)

        && (s>0)

        && (p%2 == 0)) {

        printf("Correct Values");

    } else {

        printf("Wrong Values");

    }


    return 0;

}
```

Test Data :

Input the second integer: 35

Input the third integer: 15

Input the fourth integer: 46

Expected Output:

Wrong values

20. Solve quadratic equation using Bhaskara's formula

Write a C program to print the roots of Bhaskara's formula from the given three floating numbers. Display a message if it is not possible to find the roots.

Code:

```
#include <stdio.h>

#include <math.h>

int main() {

    double a, b, c;

    double determinant;

    printf("Input the first number(a): "); scanf("%lf", &a);
    printf("Input the second number(b): "); scanf("%lf", &b);
    printf("Input the third number(c): "); scanf("%lf", &c);

    determinant = (b*b) - (4 * a * c);
```



```

    if(determinant > 0 && a!= 0){

        double x, y;

        determinant = sqrt(determinant);

        x = (-b + determinant)/(2*a);

        y = (-b - determinant)/(2*a);

        printf("Root1 = %.5lf\n",x);

        printf("Root2 = %.5lf\n",y);

    } else {

        printf("Impossible to find the roots.\n");

    }

    return 0;

}

```

Test Data :

Input the first number(a): 25

Input the second number(b): 35

Input the third number(c): 12

Expected Output:

Root1 = -0.60000

Root2 = -0.80000

21. Check integer range or error for negative > 80

Write a C program that reads an integer and checks the specified range to which it belongs. Print an error message if the number is negative and greater than 80.

Code:

```
#include <stdio.h>

int main() {

    int a;

    printf("Input an integer: ");

    scanf("%d", &a);

    if(a>=0 && a<=20) {

        printf("Range [0, 20]\n");

    }

    else if(a >= 21 && a <= 60) {

        printf("Range (21, 40]\n");

    }

    else if(a >= 41 && a <= 80) {

        printf("Range (41, 80]\n");

    } else {

        printf("Outside the range\n");

    }

}
```

```
    return 0;  
}
```

Test Data :

Input an integer: 15

Expected Output:

Range [0, 20]

22. Sum all odd numbers among five inputs

Write a C program that reads 5 numbers and sums all odd values between them.

Code:

```
#include <stdio.h>  
  
#include <stdbool.h>  
  
int is_even(int n) {  
    int flag = 0;  
    if(n % 2 == 0) {  
        flag = 1;  
    } else {  
        flag = 0;  
    }  
}
```

```
        return flag;
    }

int main()
{
    int nums[5], total = 0;

    printf("Input the first number: ");
    scanf("%d", &nums[0]);

    printf("Input the second number: ");
    scanf("%d", &nums[1]);

    printf("Input the third number: ");
    scanf("%d", &nums[2]);

    printf("Input the fourth number: ");
    scanf("%d", &nums[3]);

    printf("Input the fifth number: ");
    scanf("%d", &nums[4]);

    for(int i = 0; i < 5; i++){
        if(!is_even(nums[i])){
            total += nums[i];
        }
    }

    printf("Sum of all odd values: %d\n", total);
}
```

```
    return 0;  
}
```

Test Data :

Input the first number: 11

Input the second number: 17

Input the third number: 13

Input the fourth number: 12

Input the fifth number: 5

Expected Output:

Sum of all odd values: 46

23. Check for valid triangle and calculate perimeter

Write a C program that reads three floating-point values and checks if it is possible to make a triangle with them. Determine the perimeter of the triangle if the given values are valid.

Code:

```
#include <stdio.h>  
  
int main()  
{  
  
    float a, b, c;  
  
    printf("Input the first number: ");  
  
    scanf("%f", &a);  
  
    printf("Input the second number: ");
```

```

scanf("%f", &b);

printf("Input the third number: ");

scanf("%f", &c);

if((a+b>c) && (b+c>a) && (c+a>b) ) {

    printf("Perimeter: %.1f\n", a+b+c);

} else {

    printf("Not possible to create a triangle.");

}

return 0;

}

```

Test Data :

Input the first number: 25

Input the second number: 15

Input the third number: 35

Expected Output:

Perimeter = 75.0

24. Check if two integers are multiples

Write a C program that reads two integers and checks whether they are multiplied or not.

Code:

```

#include <stdio.h>

```

```
int main()
{
    int a, b;

    printf("Input the first number: ");
    scanf("%d", &a);

    printf("Input the second number: ");
    scanf("%d", &b);

    if(b % a == 0){
        printf("Multiplied!");
    } else {
        printf("Not multiplied.");
    }

    return 0;
}
```

Test Data :

Input the first number: 5

Input the second number: 15

Expected Output:

Multiplied!

25. Get month name from number (1–12)

Write a C program that reads an integer between 1 and 12 and prints the month of the year in English.

Code:

```
#include <stdio.h>

int main()
{
    int n;

    printf("Input a number between 1 to 12 to get month name: ");

    scanf("%d", &n);

    switch (n)
    {
        case 1: printf("January\n"); break;
        case 2: printf("February\n"); break;
        case 3: printf("March\n"); break;
        case 4: printf("April\n"); break;
        case 5: printf("May\n"); break;
        case 6: printf("June\n"); break;
        case 7: printf("July\n"); break;
        case 8: printf("August\n"); break;
```



```
case 9: printf("September\n"); break;

case 10: printf("October\n"); break;

case 11: printf("November\n"); break;

case 12: printf("December\n"); break;


default: printf("Input a number between 1 to 12.");

}


return 0;

}
```

Test Data :

Input a number between 1 to 12 to get the month name: 8

Expected Output:

August

26. Print all even numbers between 1 and 50

Write a C program that prints all even numbers between 1 and 50 (inclusive).

Code:

```
#include <stdio.h>


int main() {


    for(int i = 1; i <= 50; i++) {
```

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

            printf("%d ", i);

        }

    }

    return 0;

}
```

Test Data :

Even numbers between 1 to 50 (inclusive):

Expected Output:

2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

27. Count positive and negative numbers in 5 inputs

Write a C program that reads 5 numbers and counts the number of positive numbers and negative numbers.

Code:

```
#include <stdio.h>

int main() {

    int numbers[5];

    int pos = 0, neg = 0;

    printf("Input the first number: ");
```

```
scanf("%d", &numbers[0]);

printf("Input the second number: ");

scanf("%d", &numbers[1]);

printf("Input the third number: ");

scanf("%d", &numbers[2]);

printf("Input the fourth number: ");

scanf("%d", &numbers[3]);

printf("Input the fifth number: ");

scanf("%d", &numbers[4]);


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

    if(numbers[i] > 0){

        pos++;

    } else {

        neg++;

    }

}


printf("Number of positive numbers: %d\n", pos);

printf("Number of negative numbers: %d\n", neg);


return 0;

}
```

Test Data :

Input the first number: 5

Input the second number: -4

Input the third number: 10

Input the fourth number: 15

Input the fifth number: -1

Expected Output:

Number of positive numbers: 3

Number of negative numbers: 2

28. Count positives and calculate their average from 5 inputs

Write a C program that reads 5 numbers, counts the number of positive numbers, and prints out the average of all positive values.

Code:

```
#include <stdio.h>

int main()
{
    int numbers[5], total=0;

    int pos=0;

    float avg;

    printf("Input the first number: ");

    scanf("%d", &numbers[0]);
```

```
printf("Input the second number: ");

scanf("%d", &numbers[1]);

printf("Input the third number: ");

scanf("%d", &numbers[2]);

printf("Input the fourth number: ");

scanf("%d", &numbers[3]);

printf("Input the fifth number: ");

scanf("%d", &numbers[4]);


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

    if(numbers[i]>0)

    {

        pos++;

        total+=numbers[i];

    }

}


avg = (float)total / (float)pos;


printf("Number of positive numbers: %d\n", pos);

printf("Average value of the said positive numbers: %.2f\n", avg);

return 0;

}
```

Test Data :

Input the first number: 5

Input the second number: 8

Input the third number: 10

Input the fourth number: -5

Input the fifth number: 25

Expected Output:

Number of positive numbers: 4

Average value of the said positive numbers: 12.00

29. Sum all odd values among 5 inputs

Write a C program that read 5 numbers and sum of all odd values between them.

Code:

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

```
int is_odd(int n) {
```

```
    if(n % 2 != 0) {
```

```
        return 1;
```

```
    } else {
```

```
        return 0;
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
int numbers[5], total=0;

printf("Input the first number: ");

scanf("%d", &numbers[0]);

printf("Input the second number: ");

scanf("%d", &numbers[1]);

printf("Input the third number: ");

scanf("%d", &numbers[2]);

printf("Input the fourth number: ");

scanf("%d", &numbers[3]);

printf("Input the fifth number: ");

scanf("%d", &numbers[4]);


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

    if(is_odd(numbers[i])){

        total+=numbers[i];

    }

}


printf("Sum of all odd values: %d\n", total);

return 0;

}
```

Test Data :

Input the first number: 5

Input the second number: 7

Input the third number: 9

Input the fourth number: 10

Input the fifth number: 13

Expected Output:

Sum of all odd values: 34

30. Print squares of all even numbers up to a given value

Write a C program to find and print the square of all the even values from 1 to a specified value.

Code:

```
#include <stdio.h>

int main()
{
    printf("List of square of each one of the even values from 1 to 4:\n");

    for(int i = 1; i <= 4; i++){

        if(i%2==0){

            printf("%d^2 = %d\n", i, i*i);

        }

    }
}
```



```
    return 0;  
}
```

Test Data :

List of square of each one of the even values from 1 to a 4 :

Expected Output:

$2^2 = 4$

$4^2 = 16$

31. Check if an integer is positive/negative and even/odd

Write a C program to check whether a given integer is positive even, negative even, positive odd or negative odd. Print even if the number is 0.

Code:

```
#include <stdio.h>  
  
#include <math.h>  
  
int is_even(int n);  
  
int main()  
{  
  
    int num;  
  
    printf("Input an integer: ");  
  
    scanf("%d", &num);
```

```
if(num>0) {

    printf("Positive ");

    if(is_even(num)) printf("Even");

    else printf("Odd");

}

if(num<0)

{

    printf("Negative ");

    if(is_even(num)) printf("Even");

    else printf("Odd");

}

if(num==0) printf("Positive");

return 0;

}

int is_even(int n){

    if(n%2==0) return 1;

    else return 0;

}
```

Test Data :

Input an integer: 13

Expected Output:

Positive Odd

32. Print numbers between 1 and 100 with a specific remainder

Write a C program to print all numbers between 1 and 100 which are divided by a specified number and the remainder will be 3.

Code:

```
#include <stdio.h>

int main() {

    int n;

    printf("Input an integer: ");

    scanf("%d", &n);

    for(int i = 1; i<=100; i++){

        if(i % n == 3){

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

        }

    }

    return 0;

}
```

Test Data :

Input an integer: 25

Expected Output:

3
28
53
78

33. Find the highest value and its position from inputs

Write a C program that accepts some integers from the user and finds the highest value and the input position.

Code:

```
#include <stdio.h>

#define MAX 5

int main()
{
    int number[MAX];

    int max = 0, pos = 0;

    printf("Input 5 integers:\n");

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

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

    }

    for(int j = 0; j < MAX; j++){

        if(number[j] > max){

            max = number[j];
```

```

        pos = j+1;

    }

}

printf("Highest Value: %d\n", max);

printf("Position: %d\n", pos);

return 0;

}

```

Test Data :

Input 5 integers:

5

7

15

23

45

Expected Output:

Highest value: 45

Position: 5

34. Sum odd numbers between two given integers

Write a C program to compute the sum of consecutive odd numbers from a given pair of integers.

Code:

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

```
int main(){

    int a, b, total = 0;


    printf("Input a pair of numbers(for example 10, 2):\n");

    printf("Input first number of the pair: ");

    scanf("%d", &a);

    printf("Input second number of the pair: ");

    scanf("%d", &b);


    if(a < b) return 0;


    printf("List of odd numbers: ");

    for(int i = b; i <= a; i++){

        if(i % 2 != 0) {

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

            total += i;

        }

    }


    printf("Sum=%d\n", total);

}
```

Test Data :

Input a pair of numbers (for example 10,2):

Input first number of the pair: 10

Input second number of the pair: 2

Expected Output:

List of odd numbers: 3

5

7

9

Sum=24

35. Check if a pair of numbers is in ascending/descending order

Write a C program to check if two numbers in a pair are in ascending order or descending order.

Code:

```
#include <stdio.h>

int main()
{
    int a, b;

    printf("Input a pair of numbers (for example 10,2 : 2,10):\n");
    printf("Input first number of the pair: ");
    scanf("%d", &a);
    printf("Input second number of the pair: ");
```

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

if(a > b) {

    printf("The pair is in descending order!");

}

if(b > a) {

    printf("The pair is in ascending order!");

}

if(a==b) {

    printf("This pair has the same numbers!");

}

return 0;

}
```

Test Data :

Input a pair of numbers (for example 10,2 : 2,10):

Input first number of the pair: 10

Expected Output:

Input second number of the pair: 2

The pair is in descending order!

36. Validate a password (1234 as correct)

Write a C program to read a password until it is valid. For wrong password print "Incorrect password" and for correct password print, "Correct password" and quit the program. The correct password is 1234.

Code:

```
// validate password

#include <stdio.h>

int main()
{
    const int correctPassword = 1234;

    int inputPassword;

    int x = 1;

    while(x!=0) {

        printf("Input the password: ");

        scanf("%d", &inputPassword);

        if(inputPassword == correctPassword) {

            printf("Correct password\n");

            x = 0;

        } else {

            printf("Incorrect password. Try again!\n");

        }

    }

}
```

```
    return 0;
}
```

Test Data :

Input the password: 1234

Expected Output:

Correct password

37. Determine quadrant of Cartesian coordinates (x, y)

Write a C program to read the coordinates (x, y) (in the Cartesian system) and find the quadrant to which it belongs (Quadrant -I, Quadrant -II, Quadrant -III, Quadrant -IV).

Note: A Cartesian coordinate system is a coordinate system that specifies each point uniquely in a plane by a pair of numerical coordinates.

These are often numbered from 1st to 4th and denoted by Roman numerals: I (where the signs of the (x,y) coordinates are I(+,+), II (-,+), III (-,-), and IV (+,-).

Code:

```
// Determine quadrant of Cartesian coordinates

#include <stdio.h>

int main()
```

```
{  
  
    int x, y;  
  
    printf("Input the Coordinates(x,y): \n");  
  
    printf("x: "); scanf("%d", &x);  
  
    printf("y: "); scanf("%d", &y);  
  
    if(x>0 && y>0) printf("Quadrant-I(+,+) \n");  
  
    else if(x>0 && y<0) printf("Quadrant-II(+,-) \n");  
  
    else if(x<0 && y<0) printf("Quadrant-III(-,-) \n");  
  
    else if(x<0 && y>0) printf("Quadrant-IV(-,+) \n");  
  
    return 0;  
  
}
```

Test Data :

Input the Coordinate(x,y):

x: 25

y: 15

Expected Output:

Quadrant-I(+,+)

38. Divide two numbers or print if division isn't possible

Write a program that reads two numbers and divides the first number by the second number. If division is not possible print "Division is not possible".

Code:

```
// C basic declarations and expressions

#include <stdio.h>

int main() {

    int x, y;

    float div;

    printf("Input two numbers: \n");

    printf("x: "); scanf("%d", &x);

    printf("y: "); scanf("%d", &y);

    if(y!=0) {

        div = x / y;

        printf("%.1f\n", div);

    } else {

        printf("Division not possible.");

    }

    return 0;

}
```

Test Data :

Input two numbers:

x: 25

y: 5

Expected Output: 5.0

39. Sum all numbers between two integers, excluding multiples of 17

Write a C program to calculate the sum of all numbers not divisible by 17 between two given integer numbers.

Code:

```
// C basic declarations and expressions

#include <stdio.h>

int main() {

    int x, y;

    float div;

    printf("Input two numbers: \n");

    printf("x: "); scanf("%d", &x);

    printf("y: "); scanf("%d", &y);
```

```

    if(y!=0) {

        div = x / y;

        printf("%.1f\n", div);

    } else {

        printf("Division not possible.");

    }

    return 0;

}

```

Test Data :

Input the first integer: 50 Input the second integer: 99

Expected Output:

Sum: 3521

40. Find integers divisible by 7 with a remainder of 2 or 3

Write a C program that finds all integer numbers that divide by 7 and have a remainder of 2 or 3 between two given integers.

Code:

```

#include <stdio.h>

int main()

{

```

```
int a, b;

printf("\nInput the first integer: ");

scanf("%d", &a);

printf("\nInput the second integer: ");

scanf("%d", &b);

if(a > b) {

    int tmp = b;

    b = a;

    a = tmp;

}

for(int i = a + 1; i < b; i++){

    if(i % 7 == 2 || i % 7 == 3){

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

    }

}

return 0;

}
```

Test Data :

Input the first integer: 25

Input the second integer: 45

Expected Output:

30

31

37

38

44

41. Print n lines of 3 consecutive numbers starting from 1

Write a C program to print 3 numbers on a line, starting with 1 and printing n lines. Accept the number of lines (n, integer) from the user.

Code:

```
#include <stdio.h>

int main(){

    int r, c = 3;

    int count = 1;

    printf("Input the number of lines: ");

    scanf("%d", &r);

    for(int i = 1; i <= r; i++){

        for(int j = 1; j <= c; j++){

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



```
        count++;  
  
    }  
  
    printf("\n");  
  
}  
  
return 0;  
  
}
```

Test Data :

Input number of lines: 5

Expected Output:

1 2 3

4 5 6

7 8 9

10 11 12

13 14 15

42. Print numbers with their squares and cubes for n lines

Write a C program to print a number, its square and cube, starting with 1 and printing n lines. Accept the number of lines (n, integer) from the user.

Code:

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

```

int main() {

    int a;

    printf("Input number of lines: ");

    scanf("%d", &a);

    for(int i = 1; i<=a; i++){

        printf("%d %d %d\n", i, i*i, i*i*i);

    }

    return 0;

}

```

Test Data :

Input number of lines: 5

Expected Output:

1 1 1

2 4 8

3 9 27

4 16 64

5 25 125

43. Print p lines of q numbers in sequence starting from 1

Write a C program that reads two integers p and q, prints p number of lines in a sequence of 1 to b in a line.

Code:

```
#include <stdio.h>

int main() {

    int a, b, l;

    printf("Input number of lines: ");

    scanf("%d", &a);

    printf("Number of numbers in a line: ");

    scanf("%d", &b);

    for(int i = 1, l = 1; i <= a; i++) {

        for(int j = 1; j <= b; j++){

            printf("%d ", l);

            l++;

        }

        printf("\n");

    }

    return 0;

}
```

Test Data :

Input number of lines: 5

Number of characters in a line: 6

Expected Output:

1 2 3 4 5 6

7 8 9 10 11 12

13 14 15 16 17 18

19 20 21 22 23 24

25 26 27 28 29 30

44. Calculate the average of student math marks until termination

Write a C program to calculate the average mathematics marks of some students. Input 0 (excluding to calculate the average) or a negative value to terminate the input process.

Code:

Test Data :

Input Mathematics marks (0 to terminate): 10

15

20

25

0

Expected Output:

Average marks in Mathematics: 17.50

45. Compute series sum $S=1+1/2+1/3+...+1/50$

Write a C program to calculate the value of S where $S = 1 + 1/2 + 1/3 + \dots + 1/50$.

Code:

```
#include <stdio.h>

int main() {

    float sum = 0;

    for(int i = 1; i <= 50; i++){

        sum += (float)1/i;

    }

    printf("Value of S: %.2f\n", sum);

    return 0;

}
```

Expected Output:

Value of S: 4.50

46. Compute series sum $S=1+3/2+5/4+7/8$

Write a C program to calculate the value of S where $S = 1 + 3/2 + 5/4 + 7/8$.

Expected Output:

Value of series: 4.62

47. Find all divisors of a given integer

Write a C program that finds all the divisors of an integer.

Test Data:

Input an integer: 45

Expected Output:

All the divisor of 45 are:

1

3

5

9

15

45

48. Replace negatives and zeros in an array with 100, then print

Write a C program that reads and prints the elements of an array of length 7.

Before printing, replace every negative number, zero, with 100.

Test Data:

Input the 5 members of the array:

25

45

35

65

15

Expected Output:

Array values are:

n[0] = 25

n[1] = 45

n[2] = 35

n[3] = 65

n[4] = 15

49. Populate an array where each element is triple the previous

Write a C program to read and print the elements of an array with length 7.

Before printing, insert the triple of the previous position, starting from the second position.

For example, if the first number is 2, the array numbers must be 2, 6, 18, 54 and 162

Test Data:

Input the first number of the array: 5

Expected Output:

n[0] = 5

n[1] = 15

n[2] = 45

n[3] = 135

n[4] = 405

50. Print positions and values of elements in an array < 5

Write a C program to read an array of length 5 and print the position and value of the array elements of value less than 5.

Test Data:

Input the 5 members of the array:

15

25

4

35

40

Expected Output:

$A[2] = 4.0$