**Alaa Barka TP3**

**Exercice 1 :**

#include <stdio.h>

int main()

{

    int a, b, c;

    printf("Enter a,b and c \n");

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

    if (a > b)

    {

        if (a > c)

        {

            printf("The max is a = *%d*", a);

        }

        else

        {

            printf("The max is c = *%d*", c);

        }

    }

    else

    {

        if (b > c)

        {

            printf("The max is b = *%d*", b);

        }

        else

        {

            printf("The max is c = *%d*", c);

        }

    }

    return 0;

}

**Exercice 2 :**

#include <stdio.h>

void main()

{

    float a, b, x;

    printf("Enter a and b \n");

    scanf("*%f* *%f*", &a, &b);

    if (a == 0)

    {

        if (b == 0)

        {

            printf("Infinite solutions \n");

        }

        else

        {

            printf("There's no solution");

        }

    }

    else

    {

        x = -b / a;

        printf("The solution = *%.3f* \n", x);

    }

}

**Exercice 3.4 :**

#include <stdio.h>

int main()

{

    unsigned int x;

    printf("Enter x \n");

    scanf("*%d*", &x);

    if (x % 2 == 0)

    {

        printf("x is even \n");

    }

    else

    {

        printf("x is odd \n");

    }

    return 0;

}

**Exercice 3.12**

#include <stdio.h>

int main()

{

    unsigned int x, y, z;

    printf("Enter x,y and z \n");

    scanf("*%d* *%d* *%d*", &x, &y, &z);

*// odd or even number (x)*

    if (x % 2 == 0)

    {

        printf("x is even \n");

    }

    else

    {

        printf("x is odd \n");

    }

*// parity of two numbers*

    if ((x % 2 == 0 && y % 2 == 0) || (x % 2 != 0 && y % 2 != 0))

    {

        printf("x and y have the same parity \n");

    }

    else

    {

        printf("x and y don't have the same parity \n");

    }

    return 0;

}

**Exercice 3.15**

#include <stdio.h>

int main()

{

    unsigned int x, y, z;

    printf("Enter x,y and z \n");

    scanf("*%d* *%d* *%d*", &x, &y, &z);

*// odd or even number (x)*

    if (x % 2 == 0)

    {

        printf("x is even \n");

    }

    else

    {

        printf("x is odd \n");

    }

*// parity of two numbers*

    if ((x % 2 == 0 && y % 2 == 0) || (x % 2 != 0 && y % 2 != 0))

    {

        printf("x and y have the same parity \n");

    }

    else

    {

        printf("x and y don't have the same parity \n");

    }

*// one of the three variavbles is even*

    if (x % 2 == 0 || y % 2 == 0 || z % 2 == 0)

    {

        printf("One of the three variables is even ! \n");

    }

    else

    {

        printf("The three variables are odd ! \n");

    }

    return 0;

}

**Exercice 3.17**

#include <stdio.h>

int main()

{

    unsigned int x, y, z;

    printf("Enter x,y and z \n");

    scanf("*%d* *%d* *%d*", &x, &y, &z);

*// odd or even number (x)*

    if (x % 2 == 0)

    {

        printf("x is even \n");

    }

    else

    {

        printf("x is odd \n");

    }

*// parity of two numbers*

    if ((x % 2 == 0 && y % 2 == 0) || (x % 2 != 0 && y % 2 != 0))

    {

        printf("x and y have the same parity \n");

    }

    else

    {

        printf("x and y don't have the same parity \n");

    }

*// one of the three variavbles is even*

    if (x % 2 == 0 || y % 2 == 0 || z % 2 == 0)

    {

        printf("One of the three variables is even ! \n");

    }

    else

    {

        printf("The three variables are odd ! \n");

    }

*// two of the three variables have the same parity*

    if (x % 2 == y % 2 || x % 2 == z % 2 || y % 2 == z % 2)

    {

        printf("Two of the three variables have the same parity \n");

    }

    else

    {

        printf("Three variables don't have the same parity , which is impossible ! ? \n");

    }

    return 0;

}

**Exercice 3.20**

#include <stdio.h>

int main()

{

    unsigned int x, y, z;

    char answer;

    printf("Enter x,y and z \n");

    scanf("*%d* *%d* *%d*", &x, &y, &z);

    printf("The menu : \n a-odd or even number \n b-parity of two numbers \n c-one of the three variables is even\n other-two of the three variables have the same parity \n");

    scanf("*%s*", &answer);

    switch (answer)

    {

    case 'a':

*// odd or even number (x)*

        if (x % 2 == 0)

        {

            printf("x is even \n");

        }

        else

        {

            printf("x is odd \n");

        }

        break;

    case 'b':

*// parity of two numbers*

        if ((x % 2 == 0 && y % 2 == 0) || (x % 2 != 0 && y % 2 != 0))

        {

            printf("x and y have the same parity \n");

        }

        else

        {

            printf("x and y don't have the same parity \n");

        }

        break;

    case 'c':

*// one of the three variavbles is even*

        if ((x % 2 == 0) || (y % 2 == 0) || (z % 2 == 0))

        {

            printf("One of the three variables is even ! \n");

        }

        else

        {

            printf("The three variables are odd ! \n");

        }

        break;

    default:

*// two of the three variables have the same parity*

        if (x % 2 == y % 2 || x % 2 == z % 2 || y % 2 == z % 2)

        {

            printf("Two of the three variables have the same parity \n");

        }

        else

        {

            printf("Three variables don't have the same parity , which is impossible ! ? \n");

        }

    }

    return 0;

}

**Exercice 4:**

#include <stdio.h>

int main()

{

    float a, b;

    char operator;

    printf("Enter a and b  \n");

    scanf("*%f* *%f*", &a, &b);

    printf("Enter the operator ! \n");

    scanf("*%s*", &operator);

    switch (operator)

    {

    case '/':

        printf("*%f* *%c* *%f* = *%f*", a, operator, b, a / b);

        break;

    case '+':

        printf("*%f* *%c* *%f* = *%f*", a, operator, b, a + b);

        break;

    case '-':

        printf("*%f* *%c* *%f* = *%f*", a, operator, b, a - b);

        break;

    case '\*':

        printf("*%f* *%s* *%f* = *%f*", a, operator, b, a \* *b*);

        break;

    default:

        printf("invalid operator ! \n");

    }

    return 0;

}