UNIVERSIDAD NACIONAL DE SAN AGUSTIN DE AREQUIPA ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACION



**LABORATORIO 1**

CURSO ESTRUCTURAS DISCRETAS ii

Apellidos y Nombres: Ochoa Ortiz Sebastian Andre

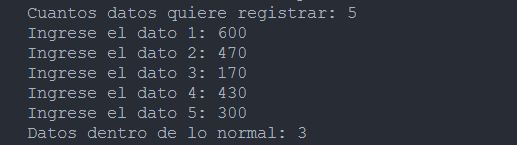
CUI: 20210694 email (Institucional): sochoao@unsa.edu.pe

Fecha: 14 de setiembre de 2022 Sección: B

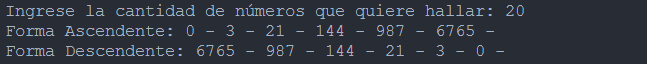
Docente: Dra. Roxana Flores Quispe / Dr. Yuber Elmer Velazco Paredes

Utilizando C++, desarrolle los siguientes ejercicios:

1. Dado un vector que registra la talla 5 perros, se pide hallar la media, la varianza, la desviación estándar y determinar cuántos animales están con una talla dentro de lo normal.
2. #include <iostream>
3. #include <vector>
4. #include <math.h>
5. int **Media**(std::vector<int> *A*)
6. {
7. int suma = 0, cantidad = 0;
8. for (int i = 0; i < *A*.**size**(); i++)
9. {
10. suma += *A***[**i**]**;
11. cantidad++;
12. }
13. return suma / cantidad;
14. }
15. int **Varianza**(std::vector<int> *A*)
16. {
17. int var = 0;
18. int cant = 0;
19. int med = **Media**(*A*);
20. for (int i = 0; i < *A*.**size**(); i++)
21. {
22. cant++;
23. var += **pow**(*A***[**i**]** - med, 2);
24. }
25. return var / cant;
26. }
27. int **Desviacion**(std::vector<int> *A*)
28. {
29. int var = **Varianza**(*A*);
30. return **sqrt**(var);
31. }
32. void **Normal**(std::vector<int> *A*)
33. {
34. int med = **Media**(*A*);
35. int des = **Desviacion**(*A*);
36. int linf = med - des;
37. int lsup = med + des;
38. int cont = 0;
39. for (int i = 0; i < *A*.**size**(); i++)
40. {
41. if (*A***[**i**]** >= linf && *A***[**i**]** <= lsup)
42. {
43. cont++;
44. }
45. }
46. std::cout **<<** "Datos dentro de lo normal: " **<<** cont **<<** std::**endl**;
47. }
48. int **main**()
49. {
50. std::vector<int> A;
51. int x = 0, n;
52. std::cout **<<** "Cuantos datos quiere registrar: ";
53. std::cin **>>** n;
54. for (int i = 0; i < n; i++)
55. {
56. std::cout **<<** "Ingrese el dato N°" **<<** i + 1 **<<** ": ";
57. std::cin **>>** x;
58. A.**push\_back**(x);
59. }
60. **Normal**(A);
61. }



1. Hallar los n primeros números de la Serie Fibonacci, almacenemos en un vector, luego en otro vector guarde solo los que son múltiplos de 3 y muéstrelos ordenados en forma ascendente y descendente.
2. #include <iostream>
3. #include <vector>
4. template <typename T>
5. void **Fibonacci**(std::vector<T> &*A*, int *N*)
6. {
7. T x = 0, y = 1, z = 1;
8. *A*.**push\_back**(x);
9. *A*.**push\_back**(y);
10. for (int i = 1; i < *N*; i++)
11. {
12. z = x + y;
13. *A*.**push\_back**(z);
14. x = y;
15. y = z;
16. }
17. }
18. template <typename T>
19. void **Print**(std::vector<T> *A*)
20. {
21. std::cout **<<** "Forma Ascendente: ";
22. for (int i = 0; i < *A*.**size**(); i++)
23. {
24. std::cout << *A*[i] << " - ";
25. }
26. std::cout **<<** std::**endl**;
27. }
28. template <typename T>
29. void **PrintDescending**(std::vector<T> *A*)
30. {
31. std::cout **<<** "Forma Descendente: ";
32. for (int i = *A*.**size**(); i > 0; i--)
33. {
34. std::cout << *A*[i-1] << " - ";
35. }
36. std::cout **<<** std::**endl**;
37. }
38. template <typename T>
39. void **Multiple3A**(std::vector<T> &*A*, std::vector<T> &*B*, int *N*)
40. {
41. **Fibonacci**(*A*, *N*);
42. for (int i = 0; i < *A*.**size**(); i++)
43. {
44. if (*A*[i] % 3 == 0)
45. {
46. *B*.**push\_back**(*A*[i]);
47. }
48. }
49. **Print**(*B*);
50. **PrintDescending**(*B*);
51. }
52. int **main**()
53. {
54. std::vector<int> A;
55. std::vector<int> B;
56. int n = 0, opc = 0;
57. std::cout **<<** "Ingrese la cantidad de n" **<<** char(163) **<<** "meros que quiere hallar: " ; std::cin **>>** n;
58. **Multiple3A**(A,B,n);
59. }



3. Crear una matriz cuadrada y mediante un Menú realizar las siguientes opciones:

a) Leer Datos.

b) Modificar Datos.

c) Mostrar Datos.

d) Mostrar la suma de la diagonal principal.

e) Mostrar la suma de la diagonal secundaria.

f) Salir

#include <iostream>

#include <vector>

void **Leer**(std::vector<std::vector<int>> &*A*)

{

    std::cout **<<** "Leyendo Datos:\n";

    for (int i = 0; i < *A*.**size**(); i++)

    {

        for (int j = 0; j < *A*.**size**(); j++)

        {

            std::cout **<<** "\tIngrese el valor [" **<<** i + 1 **<<** "] [" **<<** j + 1 **<<** "]: ";

            std::cin **>>** *A***[**i**][**j**]**;

        }

    }

}

void **Modificar**(std::vector<std::vector<int>> &*A*)

{

    std::cout **<<** "Modificando el Dato:\n";

    int fi, co, val;

    std::cout **<<** "\tIngrese la fila a modificar: ";

    std::cin **>>** fi;

    std::cout **<<** "\tIngrese la columna a modificar: ";

    std::cin **>>** co;

    std::cout **<<** "\tIngrese el valor modificado: ";

    std::cin **>>** val;

    for (int i = 0; i < *A*.**size**(); i++)

    {

        for (int j = 0; j < *A*.**size**(); j++)

        {

*A***[**fi - 1**][**co - 1**]** = val;

        }

    }

    std::cout **<<** "Dato Modificado\n";

}

void **Mostrar**(std::vector<std::vector<int>> &*A*)

{

    int valid = 0;

    for (int i = 0; i < *A*.**size**(); i++)

    {

        for (int j = 0; j < *A*.**size**(); j++)

        {

            if (*A***[**i**][**j**]** != 0)

            {

                valid++;

            }

        }

    }

    if (valid != 0)

    {

        for (int i = 0; i < *A*.**size**(); i++)

        {

            for (int j = 0; j < *A*.**size**(); j++)

            {

                std::cout **<<** *A***[**i**][**j**]** **<<** "\t";

            }

            std::cout **<<** std::**endl**;

        }

        std::cout **<<** std::**endl**;

    }

    else{

        std::cout **<<** "Matriz Vacia.\n";

    }

}

void **DiagonalP**(std::vector<std::vector<int>> *A*)

{

    int suma = 0;

    for (int i = 0; i < *A*.**size**(); i++)

    {

        suma += *A***[**i**][**i**]**;

    }

    std::cout **<<** "La Suma de la Diagonal Principal es: " **<<** suma **<<** std::**endl**;

}

void **DiagonalS**(std::vector<std::vector<int>> *A*)

{

    int suma = 0;

    for (int i = 0; i < *A*.**size**(); i++)

    {

        for (int j = 0; j < *A*.**size**(); j++)

        {

            if ((i + j) == (*A*.**size**() - 1))

                suma += *A***[**i**][**j**]**;

        }

    }

    std::cout **<<** "La Suma de la Diagonal Secundaria es: " **<<** suma **<<** std::**endl**;

}

void **Menu**()

{

    std::cout **<<** "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MEN" **<<** char(233) **<<** " MATRICES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" **<<** std::**endl**;

    std::cout **<<** "1: Leer Datos\n";

    std::cout **<<** "2: Modificar Datos\n";

    std::cout **<<** "3: Mostrar Datos\n";

    std::cout **<<** "4: Mostrar la Suma de la Diagonal Principal\n";

    std::cout **<<** "5: Mostrar la Suma de la Diagonal Secundaria\n";

    std::cout **<<** "6: SALIR\n";

}

int **main**()

{

    int fi;

    std::cout **<<** "Ingrese el tama" **<<** char(164) **<<** "o de la Matriz: ";

    std::cin **>>** fi;

    std::vector<std::vector<int>> **A**(fi, std::**vector**<int>(fi, 0));

    int opc;

    bool valid = false;

**Menu**();

    do

    {

        std::cout **<<** "Ingrese una opcion valida: ";

        std::cin **>>** opc;

        switch (opc)

        {

        case 1:

**Leer**(A);

            break;

        case 2:

**Modificar**(A);

            break;

        case 3:

**Mostrar**(A);

            break;

        case 4:

**DiagonalP**(A);

            break;

        case 5:

**DiagonalS**(A);

            break;

        case 6:

            std::cout **<<** "Cerrando Programa.\n";

            valid = true;

            break;

        default:

            std::cout **<<** "Opcion no valida.\n";

            break;

        }

    } while (valid != true);

    return 0;

}

