

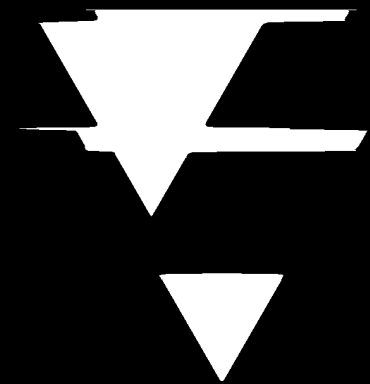
Welcome To </> CALCULADORA DE PROBABILIDADES </>

Que realice

realice una calculadora de posibilidades en la que puedes seleccionar la operacion a realizar utilizando programacion sencilla en la mayoria de los casos

el lenguaje que utilice es c++

las formulas matematicas fueron las utilizadas en la materia



Como estructurar el código

la estructura del código principal para la selección de la operación a realizar está basada en switch case en el que contamos con 11 casos diferentes de los que puedes elegir

```
switch (opcion) {  
case 1: {  
    double casosFavorables, casosPosibles;  
    std::cout << "Ingrese el numero de casos favorables: ";  
    std::cin >> casosFavorables;  
    std::cout << "Ingrese el numero de casos posibles: ";  
    std::cin >> casosPosibles;  
    double probabilidad = casosFavorables / casosPosibles;  
    std::cout << "La probabilidad simple es: " << probabilidad << std::endl;  
    break;  
}  
}
```

Como estructurar el codigo

tambien dentro de cada case declare las variables a utilizar en una salida de datos muestro al usuario los datos a requerir y tambien le pido una entrada de datos despues de ello el codigo realiza la operacion y muestra el resultado

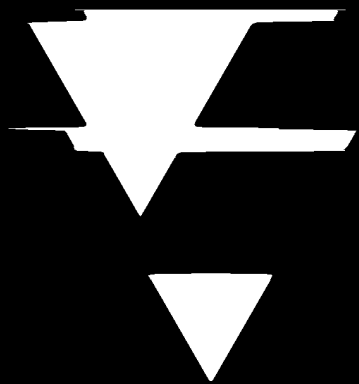
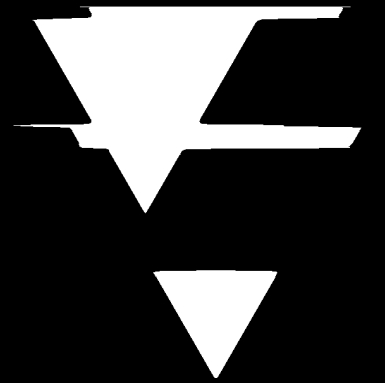
```
switch (opcion) {  
case 1: {  
    double casosFavorables, casosPosibles;  
    std::cout << "Ingrese el numero de casos favorables: ";  
    std::cin >> casosFavorables;  
    std::cout << "Ingrese el numero de casos posibles: ";  
    std::cin >> casosPosibles;  
    double probabilidad = casosFavorables / casosPosibles;  
    std::cout << "La probabilidad simple es: " << probabilidad << std::endl;  
    break;  
}  
}
```

Como realiza las Operaciones

Esto varia dependiendo el caso en el siguiente ejemplo es del caso 1 como se observa es algo relativamente muy simple en traducion a palabras

probabilidad es igual a la division de casosFavorables y casosPosibles

```
double probabilidad = casosFavorables / casosPosibles;
```



Que hace caso 1

1: Declaración de Variables:

Declara dos variables de tipo double: `casosFavorables` para almacenar el número de casos favorables y `casosPosibles` para almacenar el número total de casos posibles.

2: Entrada de Datos:

Solicita al usuario que ingrese el número de casos favorables y el número total de casos posibles. Estos valores son leídos desde la entrada estándar (generalmente el teclado) y almacenados en las variables correspondientes.

3 Cálculo de la Probabilidad:

Calcula la probabilidad dividiendo el número de casos favorables entre el número total de casos posibles. Este cálculo es posible debido a que tanto `casosFavorables` como `casosPosibles` son de tipo double, lo que garantiza que la división sea precisa y proporcione un resultado decimal.

4 Salida del Resultado:

Muestra el resultado del cálculo de la probabilidad en la salida estándar (generalmente la pantalla) mediante la función `std::cout`. La frase "La probabilidad simple es: " es seguida por el valor de la probabilidad calculada.

```
case 1: {  
    double casosFavorables, casosPosibles;  
    std::cout << "Ingrese el numero de casos favorables: ";  
    std::cin >> casosFavorables;  
    std::cout << "Ingrese el numero de casos posibles: ";  
    std::cin >> casosPosibles;  
    double probabilidad = casosFavorables / casosPosibles;  
    std::cout << "La probabilidad simple es: " << probabilidad << std::endl;  
    break;  
}
```

Que hace caso 2

Declaración de Variables:

Declara dos variables de tipo double: vecesEvento para almacenar el número de veces que ocurrió el evento y vecesTotal para almacenar el número total de veces que se realizó el experimento.

Entrada de Datos:

Solicita al usuario que ingrese el número de veces que ocurrió el evento y el número total de veces que se realizó el experimento. Estos valores son leídos desde la entrada estándar (generalmente el teclado) y almacenados en las variables correspondientes.

Cálculo de la Probabilidad:

Calcula la probabilidad experimental dividiendo el número de veces que ocurrió el evento entre el número total de veces que se realizó el experimento. Esto proporciona una estimación de la probabilidad basada en los datos observados durante el experimento.

Salida del Resultado:

Muestra el resultado del cálculo de la probabilidad en la salida estándar (generalmente la pantalla) mediante la función std::cout. La frase "La probabilidad experimental es: " es seguida por el valor de la probabilidad calculada.

```
case 2: {
    double vecesEvento, vecesTotal;
    std::cout << "Ingrese el numero de veces que ocurrio el evento: ";
    std::cin >> vecesEvento;
    std::cout << "Ingrese el numero total de veces: ";
    std::cin >> vecesTotal;
    double probabilidad = vecesEvento / vecesTotal;
    std::cout << "La probabilidad experimental es: " << probabilidad << std::endl;
    break;
}
```


Que hace caso 3

Declaración de Variables:

Se declaran dos variables de tipo double: probabilidadA para almacenar la probabilidad del evento A y probabilidadB para almacenar la probabilidad del evento B.

Entrada de Datos:

Se solicita al usuario que ingrese las probabilidades de los eventos A y B. Estas probabilidades deben estar en el rango de 0 a 1, inclusive, donde 0 representa una probabilidad nula y 1 una probabilidad total o segura.

Cálculo de la Probabilidad de Intersección:

Se calcula la probabilidad de la intersección de los eventos A y B multiplicando las probabilidades individuales de cada evento. Este cálculo se basa en la suposición de que los eventos son independientes entre sí.

Salida del Resultado:

Muestra el resultado del cálculo de la probabilidad de intersección en la salida estándar (generalmente la pantalla) mediante la función std::cout.

```
case 3: {
    double probabilidadA, probabilidadB;
    std::cout << "Recuerda Que Es En Un Rango de (1-0)\nIngrese la probabilidad de A: ";
    std::cin >> probabilidadA;
    std::cout << "Ingrese la probabilidad de B: ";
    std::cin >> probabilidadB;
    double probabilidadInterseccion = probabilidadA * probabilidadB;
    std::cout << "La probabilidad de la interseccion es: " << probabilidadInterseccion << std::endl;
    break;
}
```


Que hace caso 4

Declaración de Variables:

Se declaran tres variables de tipo double: probabilidadA para almacenar la probabilidad del evento A, probabilidadB para almacenar la probabilidad del evento B, y probabilidadInterseccion para almacenar la probabilidad de la intersección entre los eventos A y B.

Entrada de Datos:

Se solicita al usuario que ingrese las probabilidades de los eventos A y B, así como la probabilidad de la intersección entre ambos eventos.

Cálculo de la Probabilidad de la Unión:

Se calcula la probabilidad de la unión de los eventos A y B utilizando la fórmula de la probabilidad de la unión de eventos:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Donde $P(A)$ y $P(B)$ son las probabilidades de los eventos A y B respectivamente, y $P(A \cap B)$ es la probabilidad de la intersección entre los eventos A y B.

Salida del Resultado:

Muestra el resultado del cálculo de la probabilidad de la unión en la salida estándar (generalmente la pantalla) mediante la función `std::cout`.

```
case 4: {
    double probabilidadA, probabilidadB, probabilidadInterseccion;
    std::cout << "Ingrese la probabilidad de A: ";
    std::cin >> probabilidadA;
    std::cout << "Ingrese la probabilidad de B: ";
    std::cin >> probabilidadB;
    std::cout << "Ingrese la probabilidad de la interseccion de A y B: ";
    std::cin >> probabilidadInterseccion;
    double probabilidadUnion = probabilidadA + probabilidadB - probabilidadInterseccion;
    std::cout << "La probabilidad de la union es: " << probabilidadUnion << std::endl;
    break;
}
```

Que hace caso 5

Declaración de Variables

Se declaran tres variables de tipo double: probabilidadA para almacenar la probabilidad del evento A, probabilidadB para almacenar la probabilidad del evento B, y probabilidadInterseccion para almacenar la probabilidad de la intersección entre los eventos A y B.

Entrada de Datos:

Se solicita al usuario que ingrese las probabilidades de los eventos A y B, así como la probabilidad de la intersección entre ambos eventos.

Verificación de Independencia:

Se compara la probabilidad de la intersección de los eventos A y B con el producto de sus probabilidades individuales. Si la diferencia entre ambas es menor que un valor muy pequeño (generalmente el epsilon de tipo double), se considera que los eventos son independientes. De lo contrario, se consideran dependientes.

Salida del Resultado:

Muestra el resultado de la verificación en la salida estándar (generalmente la pantalla) mediante la función std::cout.

```
case 5: {
    double probabilidadA, probabilidadB, probabilidadInterseccion;
    std::cout << "Ingrese la probabilidad de A: ";
    std::cin >> probabilidadA;
    std::cout << "Ingrese la probabilidad de B: ";
    std::cin >> probabilidadB;
    std::cout << "Ingrese la probabilidad de la interseccion de A y B: ";
    std::cin >> probabilidadInterseccion;
    if (std::abs(probabilidadInterseccion - probabilidadA * probabilidadB) < std::numeric_limits<double>::epsilon()) {
        std::cout << "Los eventos A y B son independientes." << std::endl;
    }
    else {
        std::cout << "Los eventos A y B son dependientes." << std::endl;
    }
    break;
}
```

Que hace caso 6

Declaración de Variables:

Se declaran las variables necesarias para almacenar el valor esperado, la varianza, la desviación estándar y el número de valores en la distribución de probabilidad.

Entrada de Datos:

Se solicita al usuario que ingrese el número de valores en la distribución de probabilidad.

Inicialización de Vectores:

Se inicializan dos vectores para almacenar los valores y las probabilidades correspondientes.

Entrada de Valores y Probabilidades:

Se solicita al usuario que ingrese cada valor y su respectiva probabilidad.

Cálculo del Valor Esperado:

Se calcula el valor esperado multiplicando cada valor por su probabilidad y sumándolos.

Cálculo de la Varianza:

Se calcula la varianza sumando el cuadrado de las diferencias entre cada valor y el valor esperado, ponderadas por sus respectivas probabilidades.

Cálculo de la Desviación Estándar:

Se calcula la desviación estándar tomando la raíz cuadrada de la varianza.

Salida de Resultados:

Se muestra el valor esperado, la varianza y la desviación estándar en la salida estándar (generalmente la pantalla).

```
case 6: {
    double valorEsperado = 0, varianza = 0, desviacionEstandar;
    int n;
    std::cout << "Ingrese el numero de valores: ";
    std::cin >> n;
    std::vector<double> valores(n), probabilidades(n);
    for (int i = 0; i < n; i++) {
        std::cout << "Ingrese el valor " << i + 1 << ": ";
        std::cin >> valores[i];
        std::cout << "Ingrese la probabilidad de " << valores[i] << ": ";
        std::cin >> probabilidades[i];
    }
    for (int i = 0; i < n; i++) {
        valorEsperado += valores[i] * probabilidades[i];
    }
    for (int i = 0; i < n; i++) {
        double diferencia = valores[i] - valorEsperado;
        varianza += std::pow(static_cast<double>(diferencia), 2) * probabilidades[i];
    }
    desviacionEstandar = std::sqrt(varianza);
    std::cout << "El valor esperado es: " << valorEsperado << std::endl;
    std::cout << "La varianza es: " << varianza << std::endl;
    std::cout << "La desviacion estándar es: " << desviacionEstandar << std::endl;
    break;
}
```