

1: qué es un puntero y como se utiliza: un puntero es un tipo de variable que puede apuntar a otra, y que puede mostrar tanto su valor como su lugar en la memoria [por ende 5]->[5]. se utiliza mas que nada para cuestiones de memoria dinámica y acceder a otras variables de forma más fácil se expresa en c++ como: `int* variable`.

2: un arreglo es una sucesión de valores dentro de una misma variable que se acomodan dependiendo de cuantas dimensiones tenga el mismo por ejemplo: arreglo bidimensional `[] []` o arreglo tridimensional `[] [] []` cada celda es accesible a través de un número entre corchetes que `() () ()` representa la posición del mismo y se representa en c++ como: `variable [x][y][z]`. `() () ()`

3: los arreglos son usualmente de tipo entero aunque tambien se puede modificar su tipo dependiendo de tus necesidades por ejemplo un arreglo flotante: `float variable[x] [y]` o incluso de caracteres siendo como funciona el tipo `string` siendo una cadena de caracteres.

4: para obtener el tamaño de arreglo: hay varias formas de hacerlo las que conozco son las siguientes : puedes usar un apuntador o puedes usar el método que yo uso el cual es que al momento de llenar el arreglo de datos usando una variable de ciclo añadir una variable que se vaya aumentando con cada dato que se ponga en el arreglo algo así:

```
for(int i=0;i<x;i++){ // con esto simplemente tienes que después imprimir el valor de b
for(int j=0;j<y;j++){ // y tendrás el tamaño del arreglo en total para obtenerlo en que por
variable [i][j]=1; // que es es simplemente sacar la raíz cuadrada del valor de b
b=b+1
}}.
```

5: qué información nos entrega `&` en un puntero: su posición en la memoria del valor apuntado.

6: que información nos entrega `*` en un puntero: la posición de memoria del puntero.

7: diferencias en tipo booleano entero y caracter: una variable tipo booleana es una variable que simplemente se maneja en 1 y 0 siendo un 1 si y un 0 no y solo puede cambiar entre los valores previamente dichos, un entero es un numero que va del -infinito al infinito sin tener puntos decimales siendo solo números enteros y su valor es solo números enteros y el tipo caracter abarca cualquier letra y símbolos dejándole el valor de ese carácter como por ejemplo "m" es un carácter.

8: que tipo de dato puede almacenar mas de un caracter: los tipo `string` que pueden almacenar cadenas de caracteres .

9: las palabras "using namespace std;" para qué sirven en c++: sirven para poder usar las variables `cout` `cin` y parecidas que sean mas faciles de usar en vez de usar `printf` `scanf`.

10: definición palabras reservadas `if else`: `if` es una condicional la cual en caso de cumplirse mostrara una cierta porcion del codigo y `else` es una condicional que se muestra en caso de que `if` no se cumpla por ejemplo:

```
if(variable ==<>! variable 2&& y segunda condicion ||o tercera condicion)
{
```

codigo en caso de que si se cumplan las condiciones  
}else{  
codigo en caso de que no se cumplan las condiciones  
}

codigo indiferente de las condiciones.

11: cuales ciclos conoces: for este ciclo te permite tener un mayor control de el ciclo ademas de poder añadir una variable local que solo funcione dentro de su ciclo.

do, while mas sencillo con una condicion clara y mas rapido de escribir

12: escribe una sintaxis de un switch case :

```
#include <iostream>
using namespace std;
int main(){
bool a=1;
if (bool){
cout<<"a"<<endl;
}
else{
cout<<"b"<<endl;
}
}
```

13: ide: international denomination structure: estructura usada para denominar que funciona a nivel internacional... la verdad no se nunca se me quedo el dato.

14: la programación estructurada minimiza la la complejidad de los programas y reduce la cantidad de errores: verdadero.

15: fases para desarrollo de un programa: estudio del problema, plantear posibles soluciones ,elaboración de diagramas de flujo, desarrollo de algoritmos de solución.

16: la metodología de programación es un conjunto de fases o etapas orientadas a resolver un problema específico: verdadero.

17: el algoritmo que provee una información mínima es: genera

18: completa la oración: un algoritmo es una secuencia ordenada de pasos o procesos para llegar a un resultado deseado

19: tipos de algoritmos: bifurcación o decicion, repeticion o interaccion, lineales.

20: los diagramas de flujo son representaciones gráficas de los procesos que necesitas seguir para llegar a un resultado deseado usando mas que nada lógica.