1. Declare una estructura global con 6 miembros de distinto tipo y en este orden (con sus miembros inicializados):

\* int

\* char

\* float

\* char

\* double

\* int

/\* Declare otra estructura global con 6 miembros de distinto tipo y en este orden (con sus miembros inicializados):

\* int

\* int

\* float

\* char

\* char

\* double

//Declare punteros a cada una de las estructuras e imprima en pantalla el valor de sus miembros haciendo uso del operador ->Luego imprima de nuevo el valor de los miembros de la estructura pero esta vez haga uso de "type casting", operadores y aritmética de punteros. Calcule el tamaño de cada estructura con la función sizeof() y explique en un comentario sus observaciones acerca del resultado obtenido respecto a los tamaños de las estructuras

#include <iostream>

using namespace std;

struct **mi\_estructura**{

int arg1= 10;

char arg2= 'd';

float arg3= 1.0;

char arg4= 'e';

double arg5= 3.42;

int arg6= 5;

};

struct **mi\_secondstruct**{

int c1= 15;

int c2= 2008;

float c3= 1.0;

char c4='a';

char c5='b';

double c6= 4.423;

};

int **main**()

{

mi\_estructura new\_estruct;

mi\_estructura \*puntero\_1=&new\_estruct;

mi\_secondstruct mi\_estructura;

mi\_secondstruct \*puntero\_2=&mi\_estructura;

cout<<puntero\_1->arg1<<endl;

cout<<puntero\_1->arg2<<endl;

cout<<puntero\_1->arg3<<endl;

cout<<puntero\_1->arg4<<endl;

cout<<puntero\_1->arg5<<endl;

cout<<puntero\_1->arg6<<endl;

cout<<"el value int arg1: "<<\*((int\*)puntero\_1)<<endl;

cout<<"el value char arg2: "<<\*((char\*)puntero\_1+4)<<endl;

cout<<"el value float arg3:" <<\*((float\*)puntero\_1+2)<<endl;

cout<<"el value char arg4: "<<\*((char\*)puntero\_1+12)<<endl;

cout<<"el value double arg5: " <<\*((double\*)puntero\_1+2)<<endl;

cout<<"el value int arg6: "<<\*((int\*)puntero\_1+6)<<endl;

cout<<sizeof (mi\_estructura)<<"bytes"<<endl;

// Se realiza con la alineacion de bytes en este caso 4(32) en este caso el compilador reduce las variables

// float y double a 3bytes por lo que reduce el almacenamiento a 24.

cout<<puntero\_2->c1<<endl;

cout<<puntero\_2->c2<<endl;

cout<<puntero\_2->c3<<endl;

cout<<puntero\_2->c4<<endl;

cout<<puntero\_2->c5<<endl;

cout<<puntero\_2->c6<<endl;

cout<<"el value int c1: "<<\*((int\*)puntero\_2)<<endl;

cout<<"el value int c2: "<<\*((int\*)puntero\_2+1)<<endl;

cout<<"el value float c3: "<<\*((float\*)puntero\_2+2)<<endl;

cout<<"el value char c4: "<<\*((char\*)puntero\_2+12)<<endl;

cout<<"el value char c5: "<<\*((char\*)puntero\_2+13)<<endl;

cout<<"el value double c6: "<<\*((double\*)puntero\_2+2)<<endl;

cout<<sizeof (mi\_secondstruct)<<"bytes"<<endl;

// del mismo modo con la alineacion de bytes nos damos cuenta de que corresponde igualmente a 24 bytes.

return 0;

}

2. Diseñe una función llamada foo\_function() que debe tener un arreglo unidimensional del tipo y del tamanio que ud prefiera.

\* Declare un puntero al arreglo y luego agregue las líneas de código necesarias para lograr que cada vez que se haga un llamado a la función foo\_function() desde de la función main se imprima un elemento distinto del arreglo (en el orden que están en el arreglo) con ayuda de la aritmética de punteros.

int **main**()

{

int foo\_funct[10];

int \*ptrentero;

ptrentero=foo\_funct;

int i;

for (i=1;i<11;i++) {

ptrentero [i-1]=i\*i;

}

for (i=0;i<10;i++) {

cout<<ptrentero[i]<<endl;

}

return 0;

}

3. Diseñe un menú de inicio para un programa de cálculo sencillo. En el menú se debe preguntar por un número de opción

\* y cada número de opción debe ejecutar una función que hace un cálculo matemático distinto. Todas las funciones debe recibir un argumento tipo float y un valor de retorno idem. El menu debe realizarse utilizando SOLAMENTE punteros a funciones.

\* Bajo NINGUN motivo puede realizarlo utilizando sentencias "if..else if..else" o "switch..case"

#include <iostream>

using namespace std;

int **main**(){

void **func1**(float arg);

float n1,n2, m3;

void (\*puntero\_a\_funcion)(float);

float (\*puntero2)(float, float);

float **menu**(void);

{

cout<< "ingrese primer numero:"<<endl;

cin>>\*("%f",&n1);

cout<<( "ingrese segundo numero: " )<<endl;

cin>>\*("%f",&n2);

cout<<("/n +++++++++ ");

cout<<( "\n >>> OPCIONES <<< ");

cout<<( "\n 1. Sumar ");

cout<<( "\n 2. Restar ");

cout<<( "\n 3. Multiplicacion ");

cout<<( "\n Introduzca opcion ");

scanf( "%f", &puntero\_a\_funcion);

cout<<"la suma es: "<<endl;

m3=n1+n2;

cout<<"la resta es: "<<endl;

m3=n1-n2;

cout<<"la multiplicacion es: "<<endl;

m3=n1\*n2;

return 0;

}

}