

SOLUCIONES ACTIVIDAD 11NP

Diseña la estructura de datos adecuada para almacenar los datos de un producto deportivo. Para cada producto se desea almacenar:

- Código: numérico
- Tipo: puede ser: c (calzado), r (ropa) y e (material específico)
- Talla: infantil, junior y adulto
- Descuentos: puede contener un número variable de resultados, como máximo 5, ya que según el tipo de prenda y talla se aplican unos descuentos u otros. Los resultados se codificarán mediante un valor entero del 5 al 70.
- Precio: numérico.

El programa debe presentar un menú con las siguientes cuatro opciones:

1. **Introducir datos** de producto. Se pide al usuario código, tipo, talla y descuentos. El precio se deja vacío.
2. **Calcular precio** de un producto. Se pide al usuario el código de un producto, se busca entre los productos almacenados y se almacena el precio, el cual se depende del tipo, talla y del resultado del descuento.

| | Precio talla | Precio talla | Precio talla |
|---------------------|--------------|--------------|--------------|
| Tipo de producto | infantil | junior | adulto |
| Calzado | 40€ | 50€ | 60€ |
| Ropa | 65€ | 85€ | 105€ |
| Material específico | 30€ | 40€ | 50€ |

Precio = precio en función de talla y tipo* promedio descuentos/10

3. **Mostrar datos** de un producto. Se pide al usuario el código de un producto, se busca entre los productos almacenados y se muestran todos sus datos en pantalla.
4. **Salir**

```
#include <iostream>
using namespace std;
const int KMAX=100;
```

```

typedef struct{
    int cod;
    char tipo;
    char talla;
    int descuent[5];
    int numDesc;
    float precio;
}TProducto;

typedef TProducto TDeporte[KMAX];

void introduceDatos(TDeporte almacen, int &pos);
void almacenaPrecio(TDeporte almacen, int pos);
int precioBase(char tipo, char talla);
void muestraProducto(TDeporte almacen, int pos);
void escribetalla(char talla);
void escribetipo(char tipo);
char menu();

int main(){
    TDeporte almacen;
    int pos;
    char op;

    pos=0;
    do{
        op=menu();
        switch (op){
            case '1': introduceDatos(almacen, pos);
                        break;
            case '2': almacenaPrecio(almacen, pos);
                        break;
            case '3': muestraProducto(almacen, pos);
                        break;
        }
    }while(op!='4');
    return 0;
}

```

```

char menu(){
    char op;

    do{
        cout << "-----" << endl;
        cout << "1. Introducir datos del producto" << endl;
        cout << "2. Calcular precio de un producto" << endl;
        cout << "3. Mostrar datos de un producto" << endl;
        cout << "4. Salir" << endl;
        cout << "Introduce opción: ";
        cin >> op;
        if (op<'1' || op>'4')
            cout << "OPCION INCORRECTA" << endl;
    }while (op<'1' || op>'4');
    return (op);
}

void introduceDatos(TDeporte almacen, int &pos){
    int i;

    cout << "Introduce el código: ";
    cin >> almacen[pos].cod;
    do{
        cout << "Introduce el tipo(c/r/e): ";
        cin >> almacen[pos].tipo;
    }while (almacen[pos].tipo!='c' && almacen[pos].tipo!='r' &&
almacen[pos].tipo!='e');
    do{
        cout << "Introduce la talla (i/j/a): ";
        cin >> almacen[pos].talla;
    }while (almacen[pos].talla!='i' && almacen[pos].talla!='j' &&
almacen[pos].talla!='a');
    do{
        cout << "Introduce número de descuentos: ";
        cin >> almacen[pos].numDesc;
    }while(almacen[pos].numDesc<0 || almacen[pos].numDesc>5);
    for (i=0; i<almacen[pos].numDesc; i++){

```

```

do{
    cout << "Introduce descuento " << i+1 << ": ";
    cin >> almacen[pos].descuent[i];
    }while(almacen[pos].descuent[i]<5 ||
almacen[pos].descuent[i]>70);
    }
    pos++;
}

int precioBase(char tipo, char talla){
    int base;

    switch (tipo){
        case 'c': if (talla=='i')
            base=40;
            else if (talla=='j')
            base=50;
            else
            base=60;
            break;
        case 'r': if (talla=='i')
            base=65;
            else if (talla=='j')
            base=85;
            else
            base=105;
            break;
        case 'e': if (talla=='i')
            base=30;
            else if (talla=='j')
            base=40;
            else
            base=50;
            break;
    }
    return (base);
}

```

```

void almacenaPrecio(TDeporte almacen, int pos){
    int cod, i, base, j;
    bool enc;
    float sumaDesc;

    enc=false;
    sumaDesc=0.0;
    i=0;
    cout << "Introduce el código: ";
    cin >> cod;
    while (i<pos && !enc){
        if (almacen[i].cod==cod)
            enc=true;
        else
            i++;
    }
    if (enc){
        base=precioBase(almacen[i].tipo, almacen[i].talla);
        for (j=0; j<almacen[i].numDesc; j++)
            sumaDesc=sumaDesc+almacen[i].descuent[j];
        sumaDesc=sumaDesc/almacen[i].numDesc;
        almacen[i].precio=base*sumaDesc/10.0;
    }
    else
        cout << "PRODUCTO NO EXISTENTE" << endl;
}

void escribetipo(char tipo){
    cout << "Tipo: ";
    switch (tipo){
        case 'c': cout << "calzado" << endl;
            break;
        case 'r': cout << "ropa" << endl;
            break;
        case 'e': cout << "material específico" << endl;
            break;
    }
}

```

```

void escribetalla(char talla){
    cout << "Talla: ";
    switch (talla){
        case 'i': cout << "infantil" << endl;
                break;
        case 'j': cout << "junior" << endl;
                break;
        case 'a': cout << "adulto" << endl;
                break;
    }
}

```

```

void muestraProducto(TDeporte almacen, int pos){
    bool enc;
    int i, j, cod;

    enc=false;
    i=0;
    cout << "Introduce el código: ";
    cin >> cod;
    while (i<pos && !enc){
        if (almacen[i].cod==cod)
            enc=true;
        else
            i++;
    }
    if (enc){
        escribetipo (almacen[i].tipo);
        escribetalla (almacen[i].talla);
        cout << "Descuentos: ";
        for (j=0; j<almacen[i].numDesc; j++)
            cout << almacen[i].descuent[j] << " ";
        cout << endl;
        cout << "Precio: " << almacen[i].precio << endl;
    }
}

```