

Parcial Algoritmos y Programación II - 7541 Curso Calvo (2º instancia)

17 de Noviembre de 2023

Padrón:

Apellido y Nombre:

1) Indicar la salida por pantalla y escribir las sentencias necesarias para liberar correctamente la memoria.

```
int main() {  
  
    int *A, *C, *F;  
    int **B;  
    char *D, *E;  
    char G;  
    int H;  
  
    G = 'C';  
    D = &G;  
    E = D;  
    (*E) = 'B';  
    cout << (*D) << G << (*E) << endl;  
  
    H = 64 + ULTIMO DIGITO PADRON;  
    A = new int[2];  
    (*A) = H;  
    B = &A;  
    C = A + 1;  
  
    F = A;  
    F[1] = 65;  
    cout << (**B) << (*F) << (*C) << endl;  
  
    E = (char*) (F + 1);  
    G = 'D';  
    (*C) = 66;  
    cout << (*D) << (*E) << F[1] << endl;  
  
    D = (char*) (*B);  
    A[1] = 67;  
    G = 'X';  
    cout << (*E) << (*D) << (*C) << endl;  
  
    // liberar la memoria  
    // ...  
  
    return 0;  
}
```

2. Diseñar la especificación e implementar el TDA **Entrenamiento**. Un Entrenamiento se debe crear recibiendo como parámetro la cantidad de días de duración. Debe proveer operaciones para:

- Devolver la duración (en días)
- Registrar la cantidad de calorías que se consumen (ejercitar) en cada uno de los días.
- Registrar la cantidad de calorías que se incorporan por alimentos (comer) en cada uno de los días.
- Devolver el día con máxima pérdida de calorías.
- Devolver el día con mínima pérdida de calorías.
- Devolver el promedio de calorías consumidas.
- Poder estirar el entrenamiento en cantidad de días.

3. Implementar el método buscarRecetasPosibles de la clase Chef a partir de las siguientes especificaciones:

```
class Chef {  
public:  
    /* post: busca en la lista recetas aquellas Recetas que no pueden ser preparadas con los Ingredientes que se encuentran  
    * en ingredientesDisponibles. Considera que todos los Ingredientes de la Receta deben estar en  
    * ingredientesDisponibles comparando su nombre y validando que la cantidad sea la suficiente.  
    * Devuelve una nueva Pila con todos las Recetas en esta condición. */  
    Pila<Receta*> buscarRecetasPosibles(Lista<Receta*> recetas, Lista<Ingrediente*> ingredientesDisponibles);  
};
```

```
class Receta {  
public:  
  
    /* post: inicializa la Receta sin Ingredientes asignados.  
    */  
    Receta(string nombre);  
  
    /* post: elimina todos los Ingrediente de la Receta.  
    */  
    ~Receta();  
  
    /* post: devuelve todos los Ingredientes de la Receta.  
    */  
    Lista<Ingrediente*> getIngredientes();  
  
    /* post: devuelve el nombre que identifica a la Receta.  
    */  
    string getNombre();  
};
```

```
class Ingrediente {  
public:  
  
    /* post: queda inicializado con el nombre y cantidad  
    * indicados.  
    */  
    Ingrediente(string nombre, float cantidad);  
  
    ~Ingrediente();  
  
    /* post: devuelve el nombre del Ingrediente.  
    */  
    string getNombre();  
  
    /* post: devuelve la cantidad del Ingrediente.  
    */  
    float getCantidad();  
};
```

Los alumnos que tienen aprobado algún punto no deben realizarlo.

Para aprobar es necesario tener al menos el 60% de cada uno de los ejercicios correctos y completos.

Para cada método escribir pre y post condición, si recibe argumentos y cuáles, y si retorna un dato y cuál. De faltar esto, se considerará el código incompleto.

Duración del examen: 3 horas