

Padrón:

Apellido y Nombre:

Mail:

1) Indicar la salida

```
int main() {
    int *A, *C, *F;
    int **B;
    char *D, *E;
    char G;
    int H;
    H = 69;
    G = (char) H;
    A = new int;
    (*A) = H;
    F = A;
    C = F;
    (*A) = 75;
    cout << (*A) << ((*(C)-1) << (*F) << endl;
    B = &C;
    C = new int[4];
    C[1] = H;
    (**B) = H + 1;
    cout << (*A) << (*C) << (**B) << endl;
}
```

```
E = (char*) C;
D = (char*) (*B);
cout << (*E) << (*D) << endl;

(*E) = G;
cout << (*E) << (*C) << endl;

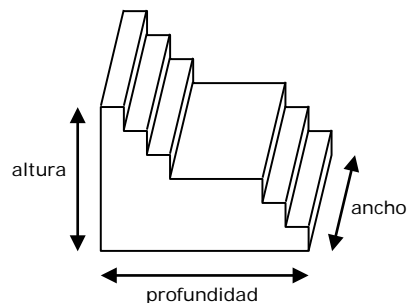
F = (int*) D;
while ( (C[1]) > 0) {
    (*C)--;
    cout << (*C) << (*D) << (**B) << endl;
    (*F) = (*F) - (**B);
    C[1] = (*F);
}
//Agregar las sentencias de delete

return 0;
```

2) Diseñar la especificación e implementar los TDAs modelen una **Escalera**, cumpliendo las siguientes características:

La **Escalera** se compone de un numero de escalones, cada escalón tiene un peso máximo. La escalera debe proveer métodos para agregar y quitar peso a un escalón, simulando que una persona lo transita.

Se requiere conocer el peso total en la **Escalera**, el peso máximo que la escalera tubo, el escalón que más peso tiene y el escalón que más peso tubo.



3) Implementar la clase FiltroDeNoticias a partir de las siguientes especificaciones:

```
class FiltroDeNoticias {
public:
    /* post: devuelve una pila de noticias de aquellas Noticias que tengan asociada TODAS las categorias dadas en conCategorias y que tengan alguna Categoria repetida en su lista.
    */
    Pila<Noticia*> buscarNoticias(Lista<Noticia*> noticias, Lista<Categoria*> conCategorias);
};
```

```
class Noticia {
public:
    /* post: inicializa la noticia con el titulo y cuerpo
    * indicados, sin categorias asociadas.
    */
    Noticia(string titulo, string cuerpo);
    /* post: devuelve el título de la noticia.
    */
    string getTitulo();
    /* post: devuelve el cuerpo de la noticia.
    */
    string getCuerpo();
    /* post: devuelve las categorias asociadas a la noticia.
    */
    Lista<Categoria*> getCategorias();
};
```

```
class Categoria {
public:
    /* post: inicializa la categoria con el nombre y la
    * descripción indicadas.
    */
    Categoria(string nombre, string descripcion);
    /* post: devuelve el nombre que la identifica.
    */
    string getNombre();
    /* post: devuelve la descripción de la categoria.
    */
    string getDescripcion();
};
```

Los alumnos que tienen aprobado el parcialito de punteros no deben realizar el ejercicio 1.

Para aprobar es necesario tener al menos el 60% 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 éste, se considerará el código incompleto.

Mantener apagados celulares, i-pod, reproductores de mp3, etc. durante el examen

Duración del examen : 2 horas y 30 minutos