



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2133 – Estructuras de Datos y Algoritmos

2024 - 2

## Tarea 0

**Fecha de entrega código:** 26 de agosto 23:59 Chile continental

**Link a generador de repos:** [CLICK AQUI](#)

### Objetivos

- Diseñar algoritmos simples en C
- Comprender las diferencias entre Arrays y Listas ligadas
- Familiarizarse con el uso de punteros y manejo de memoria

### Introducción



La leyenda del DCClub pEDDnguin comienza. . .

Nada más despertar, te das cuenta de que no estás en tu cama, sino en medio de una gran isla llena de complejidades y extrañas estructuras. Sorprendido, decides recorrer la fría isla. En medio de tu caminata, te encuentras con una sorpresa, avistas unos peculiares habitantes: pingüinos que marchan en fila, entran en orden a cualquier sitio y, en general, están obsesionados con la organización y el sistema.

Luego de visitar un par de lugares, decides ingresar a una tienda de puffles, unas pomposas criaturas que los pingüinos tienen como mascotas. Abres la puerta y, ¡sorpresa!, todo está en un claro desorden, cientos de puffles saltando en todas direcciones, lo que claramente estresa de manera excesiva a los pobres pingüinos.

Pero no te preocupes. Tú, con tus habilidades en programación y gran manejo de estructuras de datos (obviamente), te ofreces a ayudarlos, creando así lo que se llamará DCClub pEDDnguin, donde serás el encargado de dirigir todo en la tienda: desde cómo comprar puffles, intercambiarlos, manejar a los pingüinos de forma ordenada y, en general, mantener muchas otras operaciones en la tienda de mascotas.

DCClub pEDDnguin está a tu cargo, y los pingüinos confían en ti. ¡No los decepciones!

# Problema

Como persona a cargo de la tienda de mascotas, debes gestionar a los pingüinos que entran a ella. Cada pingüino tiene un inventario de puffles. Deberás implementar estructuras de datos y funciones para manejar operaciones sobre los datos de los pingüinos y sus puffles.



## Entidades

**Tienda:** Existe solamente una tienda, que tiene una capacidad máxima de ocupación de pingüinos  $K$ .

**Pingüino:** Cada pingüino tiene un ID asociado y tiene un inventario donde puede almacenar hasta  $P$  puffles, ordenados en base al momento de compra.

**Puffle:** Cada puffle tiene un ID y un color asociado.

## Parte 1: Preparando la tienda

### 1. ENTER ID N

Este evento registra a un pingüino entrando a la tienda, donde se entrega el id del pingüino, la cantidad  $N$  de puffles que lleva al inventario, y luego, en  $N$  líneas, cada puffle con su id y su color respectivamente.<sup>1</sup>

Por ejemplo, si la instrucción es

input	
1	ENTER {ID_penguin} {N}
2	{ID_puffle} {color}
3	...
4	{ID_puffle} {color}

Cada vez que ocurra este evento tu programa debe imprimir en el output anunciando el ingreso del pingüino.

output	
1	Penguin {ID_penguin} has entered with {N} puffles

En caso de que la tienda se encuentre sin capacidad para recibir al pingüino se debe entregar el siguiente output.

output	
1	Entry denied due to maximum capacity

### 2. LEAVE ID

Este evento marca la salida del pingüino ID de la tienda. Es importante destacar que este pingüino siempre estará presente dentro de la tienda.

El output debe indicar lo siguiente:

output	
1	Leaving {ID_penguin} with {N} puffles

siendo  $N$  la cantidad de puffles finales que posee el pingüino.

<sup>1</sup>Existen 10 colores de puffle: blue, red, yellow, green, black, white, pink, purple, orange, brown. Puedes mapear los colores a un ID si te resulta más fácil.

### 3. STATUS

Este evento indica el detalle de los pingüinos dentro de la tienda. Entrega el total de clientes, seguido del listado de puffles de cada pingüino<sup>2</sup>, y finalmente el total de puffles listados.

output	
1	STATUS:
2	Total penguins: X
3	Penguin {ID}:
4	Puffle {ID}
5	...
6	Puffle {ID}
7	...
8	Penguin {ID}:
9	Does not have puffles
10	...
11	Total puffles: Y

### 4. COLOR-AMOUNT COL

Este evento entrega la cantidad C de puffles de color COL que tienen los pingüinos dentro de la tienda. El output debe indicar lo siguiente:

output	
1	Total amount {COL}: {C}

## Parte 2: Trabajando con Puffles

### 1. BUY-PUFFLE ID\_penguin ID\_puffle COL

Este evento indica que un pingüino compró un puffle de color COL.

El output debe indicar lo siguiente:

output	
1	Penguin {ID_penguin} bought {ID_puffle}

En caso de que el pingüino no tenga más capacidad en su inventario para comprar el puffle se debe imprimir lo siguiente:

output	
1	Purchase denied

### 2. RUNAWAY-PUFFLE ID\_penguin ID\_puffle

Este evento indica el escape de una mascota. Se asume que la relación pingüino-puffle siempre existe en el momento dado. Una vez el puffle escapa, se considera como eliminado, es decir, no es recuperado por la tienda.

El output debe indicar lo siguiente:

output	
1	{ID_puffle} ran away from {ID_penguin}

---

<sup>2</sup>Notar que el orden del listado es en base al orden de llegada de los pingüinos a la tienda y el momento de compra de los puffles.

### 3. TRADE-PUFFLE ID\_penguin\_1 ID\_penguin\_2 ID\_puffle\_1 ID\_puffle\_2

Este evento consiste en un intercambio de puffles entre dos pingüinos. El puffle con ID ID\_puffle\_1, que pertenece al pingüino con ID ID\_penguin\_1, es intercambiado por el puffle con ID ID\_puffle\_2, que pertenece al pingüino con ID ID\_penguin\_2. Notar que al hacer el intercambio se debe respetar la posición previa en la lista de prioridades.<sup>3</sup> El output debe indicar lo siguiente:

output	
1	Exchange between {ID_penguin_1} and {ID_penguin_2}

### 4. STEAL-PUFFLE ID\_penguin\_1 ID\_penguin\_2 COLOR N\_LIMIT

Este evento consiste en que el pingüino con ID ID\_1 le **roba a lo más, los primeros N\_LIMIT** puffles de color COLOR al pingüino con ID\_penguin\_2, mostrando la cantidad robada.

Si al robar, la cantidad añadida N\_LIMIT se excede el límite  $P$  del pingüino que está robando, solo se debe robar hasta alcanzar el límite  $P$  y no superarlo.

Si la cantidad total de puffles del color indicado del pingüino a robar es menor a N\_LIMIT, se roban todos los puffles de ese color del pingüino a robar. En caso contrario, se roba N\_LIMIT. Esto siempre considerando respetar la capacidad máxima de puffles que se permite poseer.

El output debe indicar lo siguiente:

output	
1	Steal {ID_penguin_1} from {ID_penguin_2}
2	Total stolen puffles: {N}

### 5. GIVE-PUFFLES ID\_penguin\_1 ID\_penguin\_2 START END

Este evento consiste en que el pingüino con ID ID\_penguin\_1 le regala todos los puffles que se encuentren en el rango [START-END]<sup>4</sup> de su inventario al pingüino con ID ID\_penguin\_2, mostrando la cantidad regalada.

El output debe indicar lo siguiente:

output	
1	Gift from {ID_penguin_1} to {ID_penguin_2}
2	Total given away puffles: {N}

En el caso de que la cantidad total a regalar no pueda ser recibida en el inventario del pingüino que lo recibe, no se deberá regalar ningún puffle, y el output deberá ser el siguiente:

output	
1	Gift denied: {ID_penguin_2} would be full

<sup>3</sup>Si el pingüino de id\_1 intercambia su puffle con id\_1, que había llegado en segundo lugar a su posesión, con el puffle con id\_2 del pingüino de id\_2, que estaba en la novena posición en su lista, después del intercambio, el puffle con id\_2 pasará a ocupar la segunda posición en la lista del pingüino id\_1, mientras que el puffle con id\_1 ocupará la novena posición en la lista del pingüino id\_2

<sup>4</sup>Notar que este rango es en relación al orden de llegada de los puffles y no en relación a sus ID. En otras palabras, es un rango de índices del inventario.

## Consideraciones Generales

- En los output no se usan caracteres especiales, como por ejemplo ü.
- La cantidad de puffles con las que entra un pingüino es siempre menor o igual a  $P$ .
- Una vez un pingüino sale de la tienda no vuelve a entrar.
- La tienda puede vender infinitos puffles.
- Para obtener un output correcto es necesario respetar los espacios adecuados, considerando tanto entre las palabras como los márgenes de impresión.
- Los ID son únicos tanto para pingüinos como para puffles.
- Un mismo puffle no puede ser propiedad de más de un pingüino.
- En caso que un pingüino no pueda entrar dado que la tienda esté llena, igual se entregarán los datos de los puffles pertenecientes a este pingüino.
- Al entregar IDs en los eventos se puede asumir que estas siempre serán válidas.

## Ejecución

Tu programa se debe compilar con el comando **make** y debe generar un ejecutable de nombre **puffle** que se ejecuta con el siguiente comando:

```
./puffle input output
```

donde **input** será un archivo con los eventos a simular y **output** el archivo donde se guardarán los resultados.

En caso de querer correr el programa para ver los leaks de memoria utilizando **valgrind** deberás utilizar el siguiente comando:

```
valgrind ./puffle input output
```

Tu tarea será ejecutada con archivos de creciente dificultad, asignando puntaje a cada una de estas ejecuciones que tenga un output igual al esperado. A continuación detallaremos los archivos de input y output.

## Input

El archivo de input comenzará con el número **K** que indica la capacidad de la tienda, seguido del número máximo de puffles en el inventario de los pingüinos **P**, seguido de un número **E** que corresponde al número de eventos a recibir, y las **E** líneas correspondientes a cada uno.<sup>5</sup>

input	
1	2
2	3
3	11
4	ENTER 1 3
5	4 blue
6	7 green
7	2 black
8	ENTER 3 1
9	1 blue
10	ENTER 2 0
11	STATUS
12	BUY-PUFFLE 1 10 yellow
13	BUY-PUFFLE 3 9 red
14	TRADE-PUFFLE 1 3 7 1
15	STEAL-PUFFLE 1 3 green 3
16	RUNAWAY-PUFFLE 3 7
17	GIVE-PUFFLES 1 3 0 1
18	STATUS

---

<sup>5</sup>Notar que **E** indica solamente el numero de eventos, por ende, en tu solución debes considerar los múltiples inputs del evento **ENTER**.

## Output

El output deberá consistir de un archivo con la información solicitada por cada evento en el archivo de input. Ojo, ten en consideración los saltos de línea (`\n`), donde todas las líneas terminan con un salto. Así también, considera los espacios de cada impresión para que tu output coincida con el de los casos de prueba.

output	
1	Penguin 1 has entered with 3 puffles
2	Penguin 3 has entered with 1 puffles
3	Entry denied due to maximum capacity
4	STATUS:
5	Total penguins: 2
6	Penguin 1:
7	Puffle 4
8	Puffle 7
9	Puffle 2
10	Penguin 3:
11	Puffle 1
12	Total puffles: 4
13	Purchase denied
14	Penguin 3 bought 9
15	Exchange between 1 and 3
16	Steal 1 from 3
17	Total stolen puffles: 0
18	7 ran away from 3
19	Gift from 1 to 3
20	Total given away puffles: 2
21	STATUS:
22	Total penguins: 2
23	Penguin 1:
24	Puffle 2
25	Penguin 3:
26	Puffle 9
27	Puffle 4
28	Puffle 1
29	Total puffles: 4

## Evaluación

La nota de tu tarea es calculada a partir de testcases de Input/Output. La ponderación se descompone de la siguiente forma:

- 90 % a la nota entre las partes 1 y 2
  - 25 % a la nota de los tests de la parte 1
  - 75 % a la nota de los tests de la parte 2
- 10 % Por manejo de memoria
  - 5 % Por no poseer leaks de memoria
  - 5 % Por no poseer errores de memoria

Para cada test de evaluación, tu programa deberá entregar el output correcto en **menos de 5 segundos** y utilizar menos de 1 GB de RAM<sup>6</sup>. De lo contrario, recibirás 0 puntos en ese test.

## Recomendación de tus ayudantes

Recomendamos fuertemente que utilices esta tarea para aprender bien C. Antes de partir la tarea, recomendamos que programes algunos pequeños programas empleando lo que aprendas en las ayudantías y cápsulas de C. Recuerda compilar el código como se indica en la tarea, y de preguntar cualquier duda o por problemas que te surjan en [Discussions](#).

## Entrega

**Código:** GIT - Rama principal del repositorio asignado, que debes generar con el link dado al principio de este enunciado. Se entrega a más tardar el día de entrega a las 23:59 hora de Chile continental.

**Atraso:** Esta tarea NO considera la política de atraso y cupones [especificada en el repositorio del curso](#).

## Integridad académica

Este curso se adscribe al Código de Honor establecido por la Escuela de Ingeniería. Todo trabajo evaluado en este curso debe ser hecho **individualmente** por el alumno y **sin apoyo de terceros**. Se espera que los alumnos mantengan altos estándares de honestidad académica, acorde al Código de Honor de la Universidad. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Pregrado de la Escuela de Ingeniería.

---

<sup>6</sup>Puedes revisarlo con el comando `htop` u ocupando `valgrind --tool=massif`