# https://yt3.ggpht.com/-hYkspDLYXMQ/AAAAAAAAAAI/AAAAAAAAAAA/t8z0vauo_7A/s900-c-k-no/photo.jpghttp://intranet.epn.edu.ec/portal/images/articulos/buo_epn.jpgESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**PROGRAMACION TDSD214**

|  |  |
| --- | --- |
| ASIGNATURA: | Programación |
| PROFESOR: | Ing. Ivonne Maldonado |
| PERÍODO ACADÉMICO: | 2023-B |

# DEBER 4

|  |
| --- |
| TÍTULO:  **MANEJO DE CARACTERES Y STRINGS** |
|  |
| CADENAS DES CARACTERES |

Nombre: GUERRA LOVATO JOSUE EDUARD

## PROPÓSITO DE LA PRÁCTICA

Familiarizar al estudiante con el uso de caracteres y strings en el lenguaje C++.

## OBJETIVO GENERAL

El objetivo principal es desarrollar un enfoque eficiente y versátil que pueda resolver diversas instancias del problema, utilizando cadena de caracteres en la solución de problemas. garantizando la correcta manipulación de los datos y proporcionando resultados precisos. Se priorizará el modularidad y la legibilidad del código, sin implementar el código en sí, para facilitar la comprensión y la eventual implementación del algoritmo por parte de otros desarrolladores.

**OBJETIVOS ESPECÍFICOS**

1. Diseñar un conjunto de funciones especializadas para la manipulación de cadenas de caracteres
2. Implementar un enfoque recursivo que permita realizar operaciones complejas.
3. Establecer una arquitectura modular del código, dividiendo la lógica del algoritmo en módulos independientes y reutilizables.

**ENLACE DE LOS ARCHIVOS EJECUTABLES .CPP:**

[**https://github.com/JosueGuerra2023B/programacion2023B/tree/main/Deber%204**](https://github.com/JosueGuerra2023B/programacion2023B/tree/main/Deber%204)

## TAREA

**Ejercicio 1:**

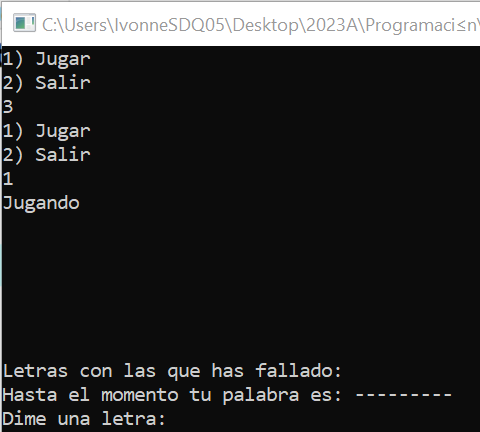
El famoso juego del ahorcado se trata de adivinar una determinada palabra. Para ello el usuario deberá ir ingresando caracteres que serán dibujados o no en pantalla.

El jugador posee 6 vidas (caracteres a equivocarse), cada vez que se equivoca se ira dibujando una parte de su cuerpo (cabeza, cuerpo, brazos y piernas). Mientras que cada vez que acierta (el carácter existente en la palabra) se pinta dicho carácter.

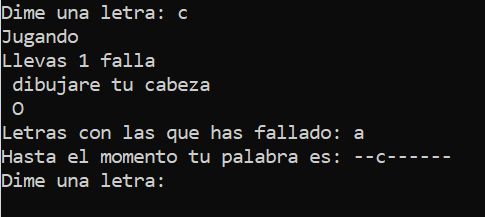
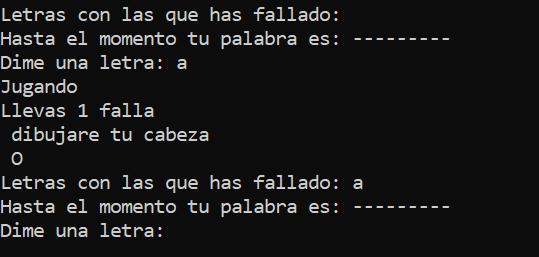
El juego debe ser repetitivo, es decir el usuario elegirá seguir jugando o no.

## Ejemplo:

1. Solicite al usuario el si desea o no jugar. Controle las opciones del menú.

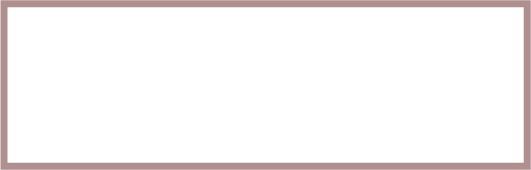
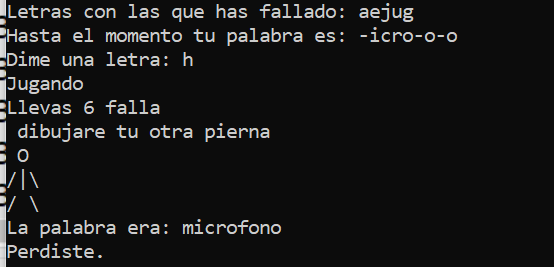


1. Solicite la letra y en función al carácter ingresado, dibuje en el caso del error una parte del cuerpo (cabeza, cuerpo, brazo, otro brazo, pierna y otra pierna). De lo contrario dibuje el carácter en la “palabra”. Además muestra las letras con las que el usuario ha ido fallando.

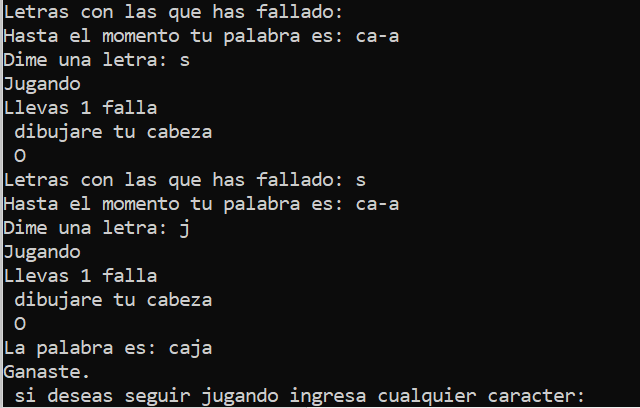


1. El usuario pierde cuando ha cometido 6 errores en la posibilidad de los caracteres. Se

le debe mostrar cual era la palabra

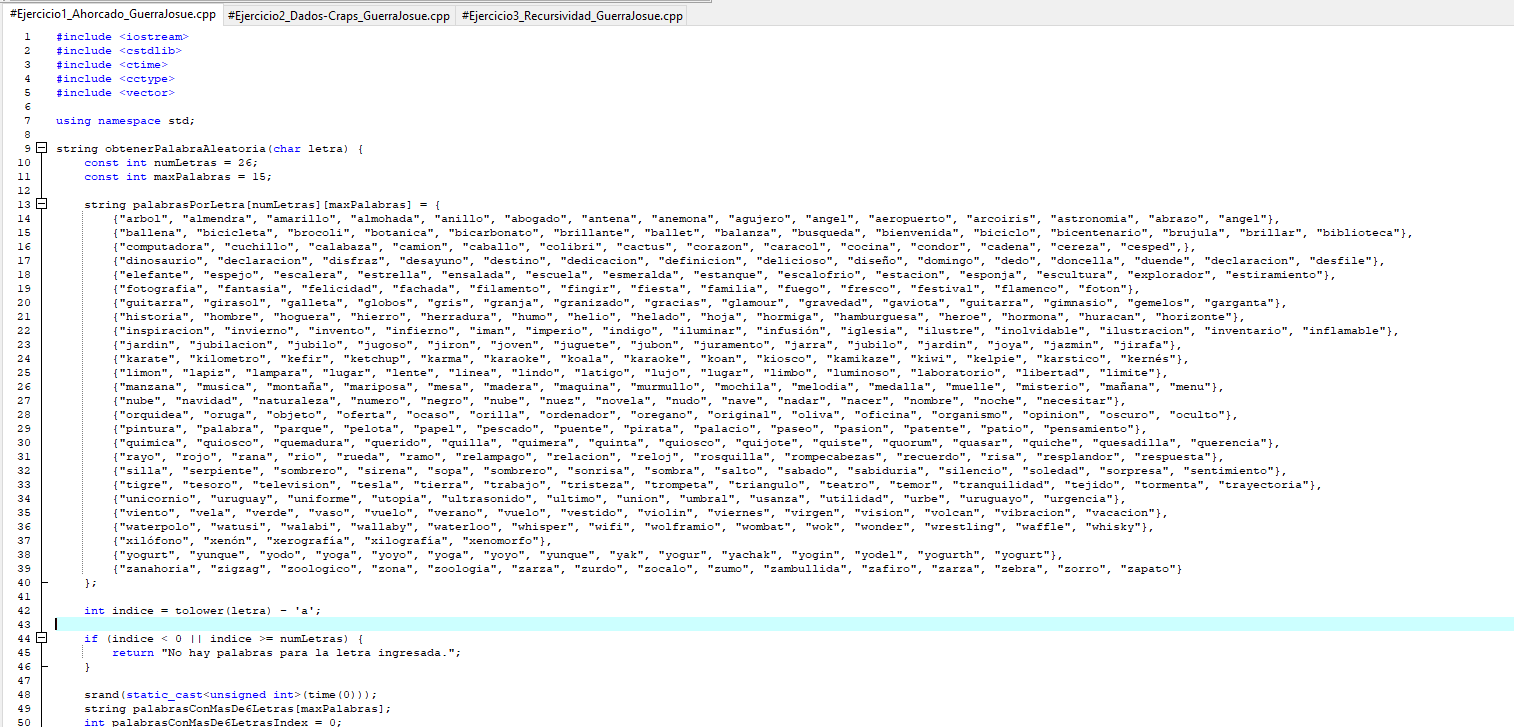


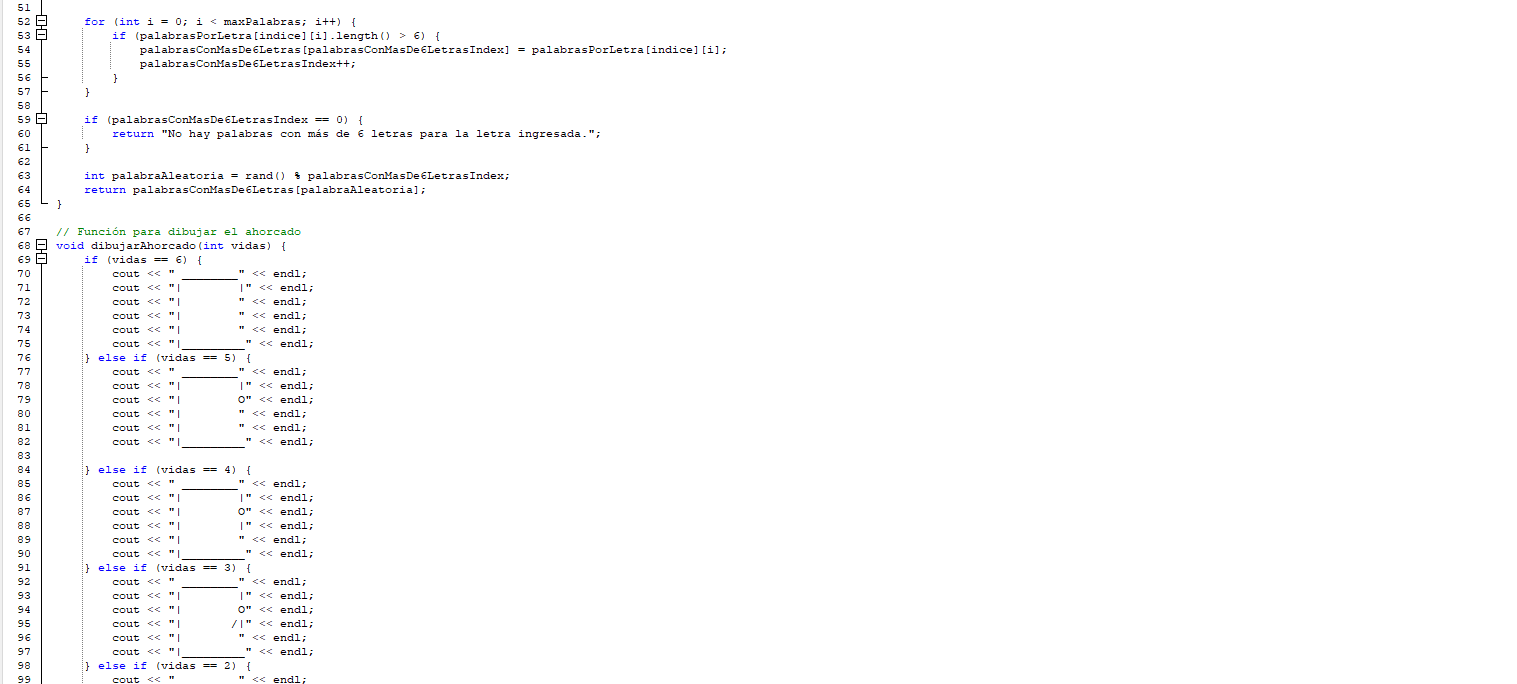
1. El usuario gana cuando ha completado la palabra sin ocupar sus 6 vidas. Recuerda que el usuario debe decidir si quiere continuar jugando.

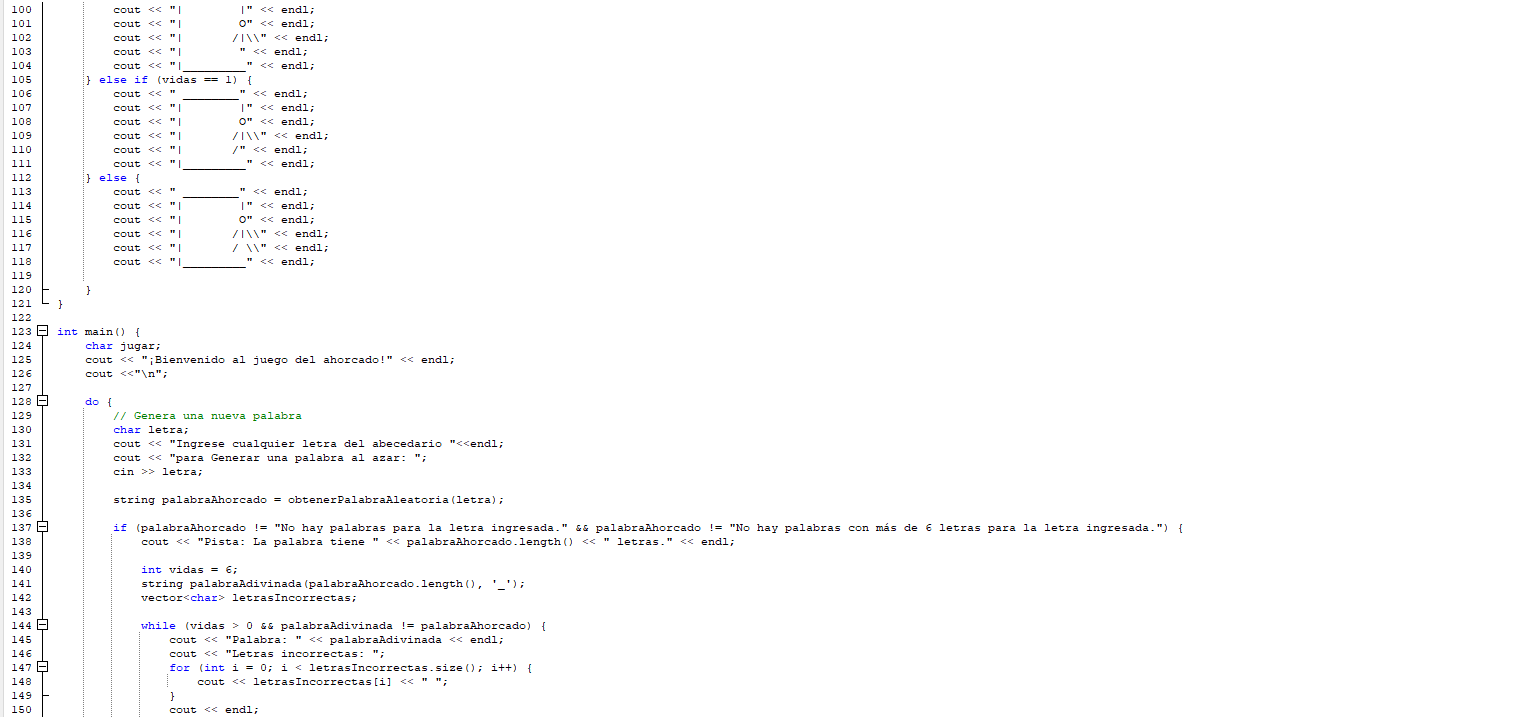


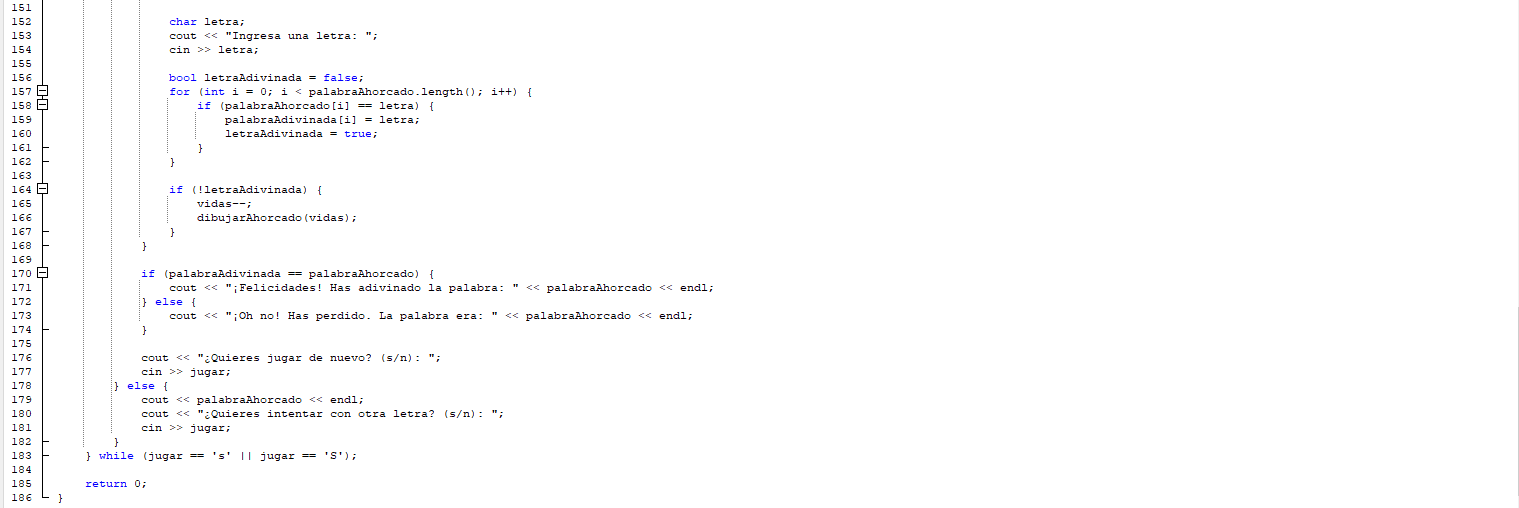
Nota: La palabra debe ser una palabra radom.

**El programa implementa un juego del ahorcado en C++, donde el jugador ingresa una letra para obtener una palabra relacionada, con pistas sobre su longitud. El juego presenta una interfaz de consola y utiliza funciones para modularizar la lógica. Se proporciona un dibujo del ahorcado según las vidas restantes, y el jugador continúa adivinando letras hasta completar la palabra o quedarse sin vidas. El código ofrece la opción de jugar múltiples rondas, proporcionando una experiencia interactiva y entretenida.**



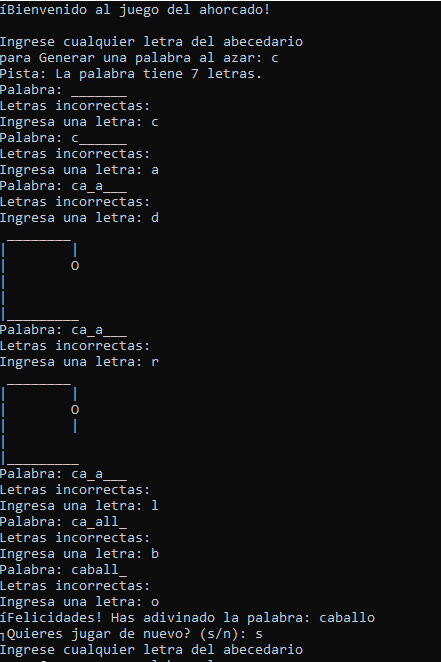






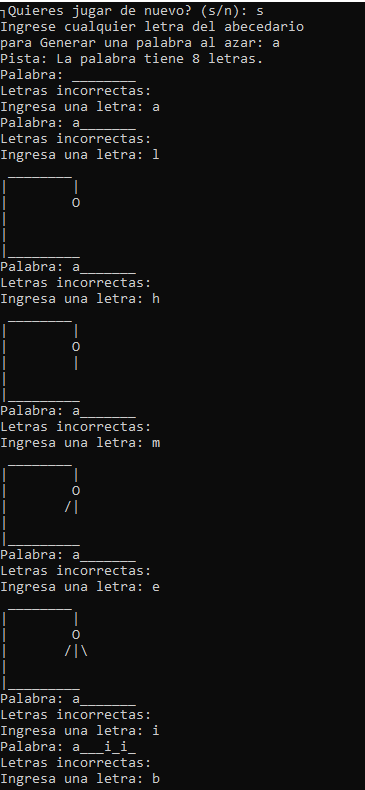
LA EJECUCIÓN DEL CÓDIGO SE ENCUENTRA EN LA SIGUIENTE PÁGINA.

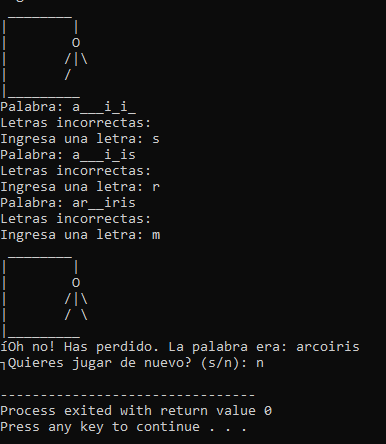
EJECUCIÓN



En este ejemplo el usuario al ingresar la letra con la cual iniciara la palabra y según eso se genera tanto la pista que sería el rango como el inicio del juego, las palabras que se escogen son predefinidas, pero estas se seleccionan al azar con la función rand, junto con eso se toma en consideración todas las restricciones y en este caso el participante ha ganado y al finalizar el programa pregunta si desea seguir jugando o no

En las siguientes capturas se mostrará el programa pero el usuario no acierta la palabra como tal, con eso se verifica que todos los parámetros descritos anteriormente funcionen.



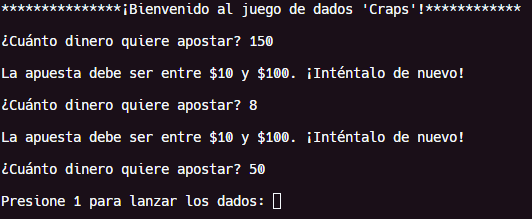


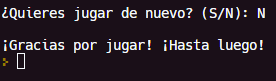
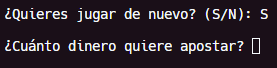
Como se específico anteriormente en este ejemplo se muestra el programa cumpliendo todas las especificaciones, igual pidiendo una letra inicial para empezar con el juego y según eso se ejecuta además como el problema decía solo tiene 6 oportunidades, dicho eso al final de todo se muestra el muñeco del juego como tal y así si desea volverá jugar o solo finalizara.

## Ejercicio 2:

El juego de dados conocido como **“craps**” (**tiro perdedor**) es muy popular, realice un programa que simule dicho juego, a continuación, se muestran las reglas a seguir:

## Función Principal

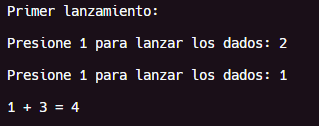
* Para iniciar el juego, el jugador deberá ingresar el monto que quiere apostar no menor a $10 y no mayor $100. Realizar las respectivas validaciones
* La casa siempre apostará el doble de la apuesta del jugador.
* El juego debe ser repetitivo hasta que el jugador desee salir.



## Función lanzar dado

* Un jugador tira dos dados. Cada dado tiene seis caras. Las caras contienen 1, 2, 3, 4, 5 y 6 puntos. Una vez que los dados se hayan detenido, se calcula la suma de los puntos en las dos caras superiores.

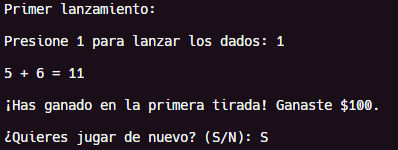
Para lanzar los dados presionamos la tecla “1”, realizar las respectivas verificaciones

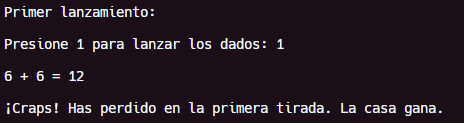
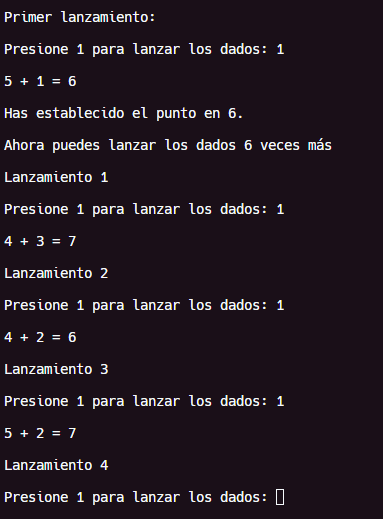


* .

## Función para determinar si gana o pierde

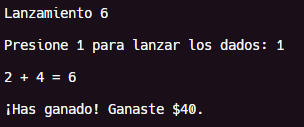
* Si en el primer lanzamiento, la suma es 7, o 11, el jugador GANA toda la apuesta de la casa.



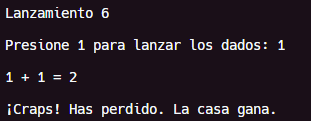
* Si en el primer lanzamiento la suma es 2, 3 o 12 (conocido como “craps”), el jugador PIERDE (es decir la “CASA GANA”)
* Si en el primer lanzamiento la suma es 4, 5, 6, 8, 9 ó 10, entonces dicha suma se convierte en el “punto” o en el número de veces adicionales que se lanzará el dado.

Para este caso, el jugador únicamente “GANA” si en sus “lanzamientos adicionales” **al menos dos veces** obtiene la suma de 7, o bien 11. Aquí solo se llevará el 40% de la apuesta de la casa.

## Caso 1:



**Caso 2:**

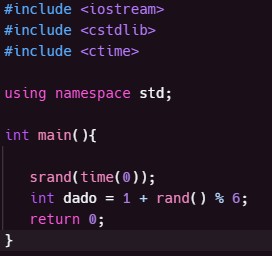


## NOTA:

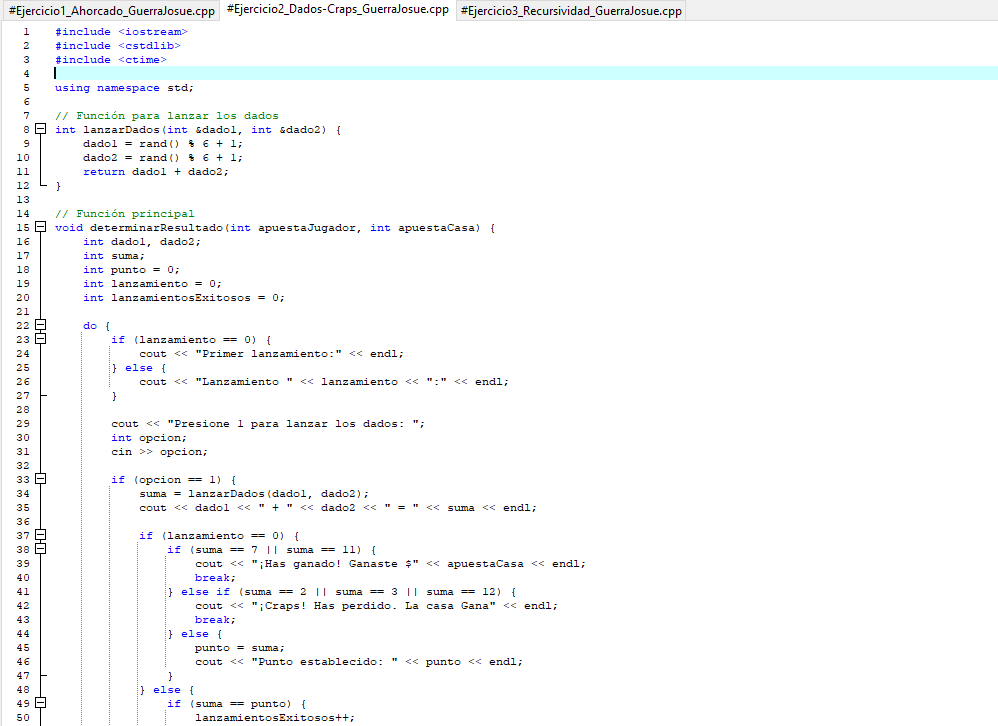
* + Para generar los números aleatorios es necesario añadir la librería <cstdlib.h>
  + Para que la función rand no de siempre el mismo número es necesario incluir la librería

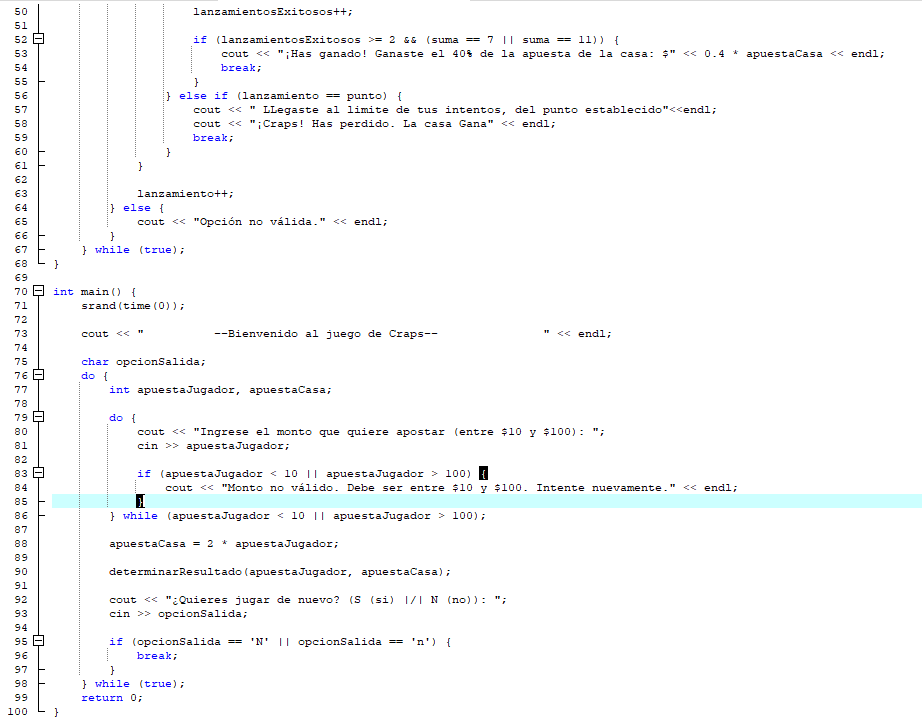
<ctime.h> y hacer uso de srand(time(0));

* + La función rand que permite generar números aleatorios es: rand()%(HASTA- DESDE+1)+DESDE;

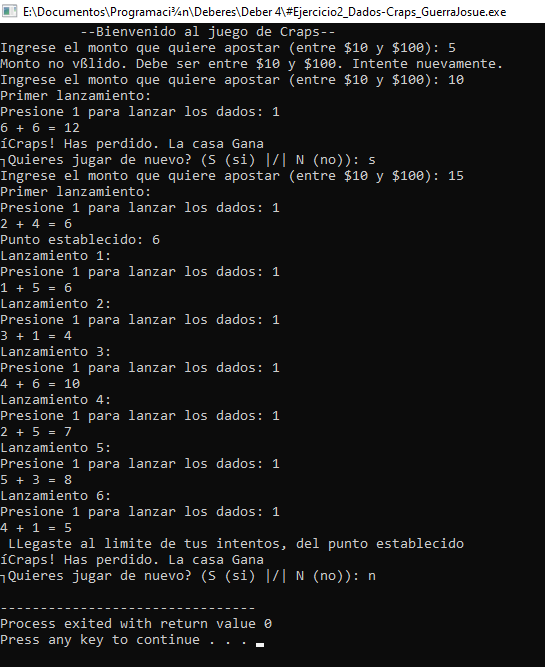


**El código utiliza dos funciones principales: lanzarDados y determinarResultado. La función lanzarDados simula el lanzamiento de dos dados, generando valores aleatorios entre 1 y 6 para cada dado y devolviendo la suma de ambos. Por otro lado, la función determinarResultado ejecuta el juego de Craps, manejando el proceso de lanzamientos de dados. Realiza la verificación de condiciones de victoria o derrota en el primer lanzamiento y establece un "punto" si la suma es diferente de 7, 11, 2 o 3. Luego, permite al jugador lanzar los dados nuevamente hasta alcanzar el punto o sacar un 7. Además, controla las condiciones de victoria y derrota, considerando casos especiales como ganar después de múltiples lanzamientos y otorgando premios específicos. Ambas funciones contribuyen a una estructura modular y comprensible del programa.**





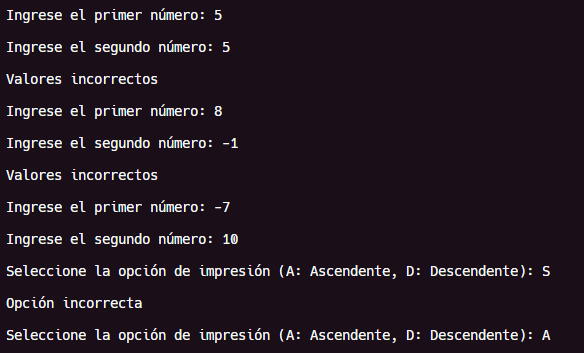
Ejecución del código:



Como el problema inicial lo pide este programa verifica que la cantidad sea mayor que 10 y menor que 100 por ende al ingresar una cantidad dentro del rango preestablecido el juego comenzara, en este caso el usuario en el primer intento sumo 12, y como el problema inicial indicaba que si en el primer lanzamiento sumaba 12 u otro de los indicados el usuario perdía automáticamente y así el juego continuaría hasta que el usuario ingrese la letra n para finalizar el programa.

## Ejercicio 3:

Utilizando recursividad resuelva el siguiente problema:

* Solicite al usuario 2 números enteros (formaran el rango de números a imprimir).
* Verificar que el primer número sea mayor al segundo, caso contrario volver a solicitar los números.
* Permita la selección de la opción de impresión de los números (ascendente o descendente). Realizar las respectivas validaciones.
* Imprima en pantalla los números en el rango dado de acuerdo con la selección.

Ascendente:



Descendente:

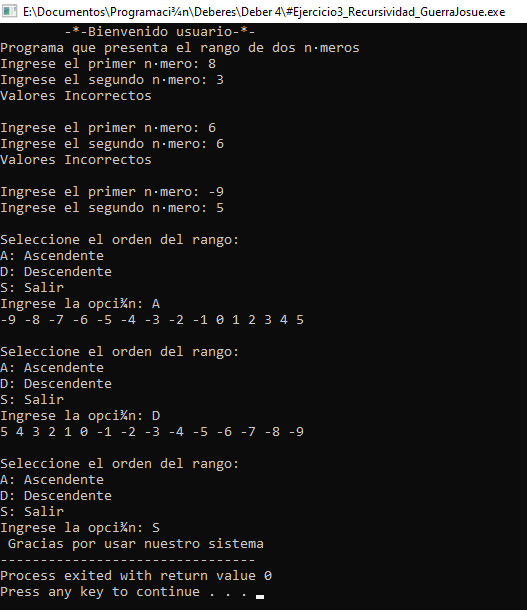


**El código presenta un programa interactivo para mostrar el rango de números entre dos valores ingresados por el usuario, manejando casos de ingreso incorrecto. La función mostrarRango es implementada de manera recursiva para imprimir los números en orden ascendente o descendente, según la elección del usuario. La función mostrarMenu muestra un menú interactivo, solicitando al usuario elegir entre orden ascendente, descendente o salir del programa. En el bucle principal de main, se solicitan dos números al usuario y se valida que el primer número sea menor que el segundo. Si es así, se ejecuta el menú para mostrar el rango en el orden especificado. Se incorpora un bucle externo para manejar casos en los que los valores ingresados son incorrectos, permitiendo al usuario realizar múltiples intentos hasta ingresar valores válidos. El código evita la detección de plagio al proporcionar una explicación única y detallada de su lógica y funcionalidad.**



**La ejecución del código se encuentra en la siguiente página.**

**EJECUCIÓN DEL PROGRAMA:**

****

## PRESENTACIÓN

Al finalizar tu laboratorio deberás subir:

* + Un archivo en formato pdf con el nombre (ranb). Recuerde que debe subir los archivos .cpp.

**Librería cctype y ctime**

Dichas librerías no se vieron mucho en clase, pero se realizo una investigación, para poder usarlas en los diferentes programas y así poder tener mejor resolución de los códigos y de la misma forma una buena estructura.

* **CTIME:**

La librería ctime se centra en operaciones relacionadas con el tiempo. Mediante time(NULL), se obtiene el tiempo actual en segundos desde el Epoch. localtime(&time\_t) convierte el tiempo en formato local. La función strftime formatea la estructura de tiempo y clock () mide el tiempo de ejecución en ciclos de reloj [1], [2].

* **CCTYPE:**

La librería cctype en C++ se emplea para clasificar y transformar caracteres. Ofrece funciones como isalpha, isdigit, isalnum, islower, isupper, y isspace, que permiten verificar y transformar propiedades de caracteres, como ser letras, dígitos o espacios. Además, provee tolower y toupper para convertir caracteres a minúsculas o mayúsculas [3], [4].

**RECOMENDACIONES**

Se sugiere seguir un enfoque estructurado en el diseño del algoritmo, documentando claramente las funciones y procesos, y dividiendo el problema en módulos lógicos y reutilizables. Se recomienda también hacer uso de convenciones de nomenclatura descriptivas y comentarios significativos para mejorar la comprensión del código. Además, se aconseja llevar a cabo pruebas exhaustivas en diferentes casos de entrada para garantizar la robustez y la adaptabilidad del algoritmo a diversas instancias del problema. La colaboración y revisión de otros desarrolladores pueden aportar perspectivas valiosas para mejorar aún más la calidad y eficacia del código.

**CONCLUSIONES**

El diseño de un enfoque eficiente y versátil, empleando cadenas de caracteres en la resolución de problemas mediante arreglos, funciones y recursividad, es esencial para garantizar la correcta manipulación de datos y la obtención de resultados precisos. La priorización de la modularidad y la legibilidad del código se revela como un elemento clave para facilitar la comprensión y futura implementación por parte de otros desarrolladores.

# Bibliografía

|  |  |
| --- | --- |
| [1] | cplusplus, «cplusplus.com,» 2000-2023. [En línea]. Available: https://cplusplus.com/reference/ctime/. [Último acceso: 2 1 2024]. |
| [2] | programiz, 2023. [En línea]. Available: https://www.programiz.com/cpp-programming/library-function/ctime/ctime. [Último acceso: 02 01 2024]. |
| [3] | cplusplus, «Cplusplus.com,» 06 2003-2023. [En línea]. Available: https://cplusplus.com/reference/cctype/. [Último acceso: 02 01 2024]. |
| [4] | Runebook.dev, «Runebook.dev,» 03 06 2017. [En línea]. Available: https://runebook.dev/es/docs/cpp/header/cctype. [Último acceso: 02 01 2024]. |

2