# 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

Utilizar cadena de caracteres en la solución de problemas.

## TAREA

**Ejercicio 1:**

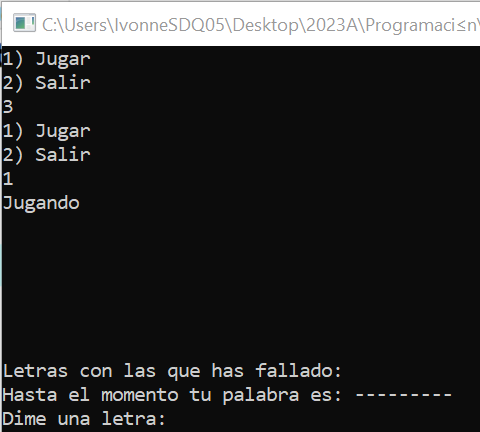
El famoso juego del ahorcado se trata de adivinar una determinada palabra. Para ello el usuario deberá ir ingresando caracteres que serán dibujados o no en pantalla.

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.

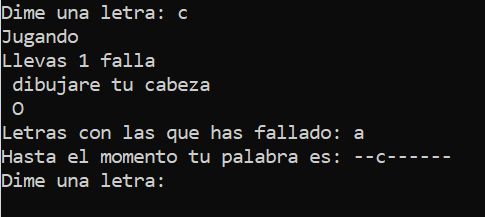
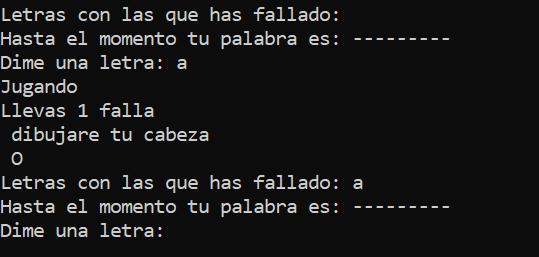
El juego debe ser repetitivo, es decir el usuario elegirá seguir jugando o no.

## Ejemplo:

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

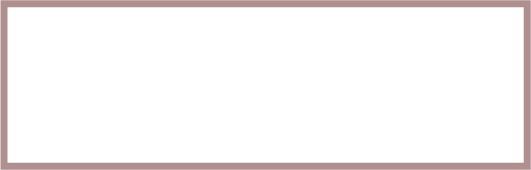
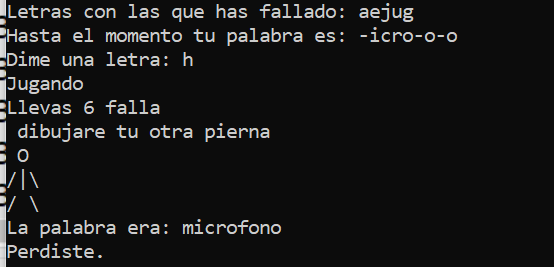


1. Solicite la letra y en función al carácter ingresado, dibuje en el caso del error una parte del cuerpo (cabeza, cuerpo, brazo, otro brazo, pierna y otra pierna). De lo contrario dibuje el carácter en la “palabra”. Además muestra las letras con las que el usuario ha ido fallando.

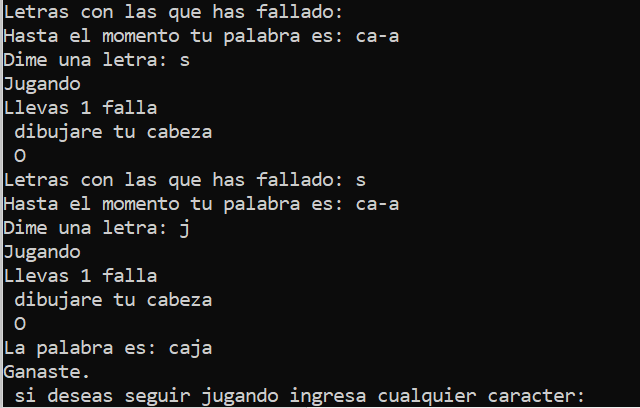


1. El usuario pierde cuando ha cometido 6 errores en la posibilidad de los caracteres. Se

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.



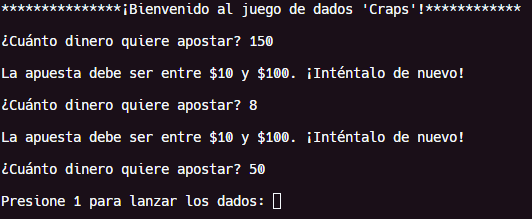
Nota: La palabra debe ser una palabra radom.

## Ejercicio 2:

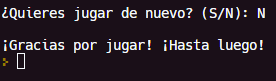
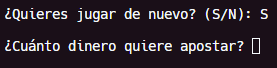
El juego de dados conocido como **“craps**” (**tiro perdedor**) es muy popular, realice un programa que simule dicho juego, a continuación, se muestran las reglas a seguir:

## 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.

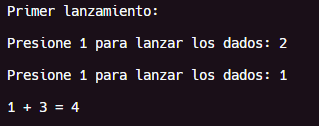


* La casa siempre apostará el doble de la apuesta del jugador.
* El juego debe ser repetitivo hasta que el jugador desee salir.



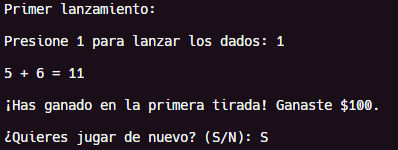
## Función lanzar dado

* 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.

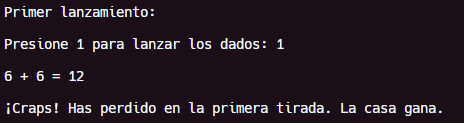


## Función para determinar si gana o pierde

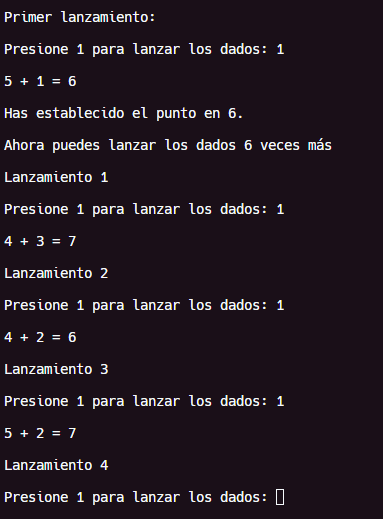
* Si en el primer lanzamiento, la suma es 7, o 11, el jugador GANA toda la apuesta de la casa.



* Si en el primer lanzamiento la suma es 2, 3 o 12 (conocido como “craps”), el jugador PIERDE (es decir la “CASA GANA”).

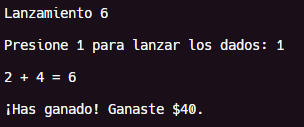


* Si en el primer lanzamiento la suma es 4, 5, 6, 8, 9 ó 10, entonces dicha suma se convierte en el “punto” o en el número de veces adicionales que se lanzará el dado.

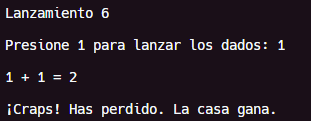


Para este caso, el jugador únicamente “GANA” si en sus “lanzamientos adicionales” **al menos dos veces** obtiene la suma de 7, o bien 11. Aquí solo se llevará el 40% de la apuesta de la casa.

## Caso 1:



**Caso 2:**

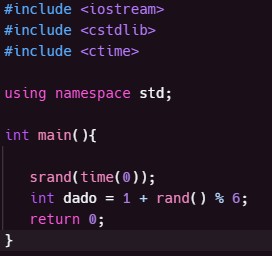


## NOTA:

* + 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));

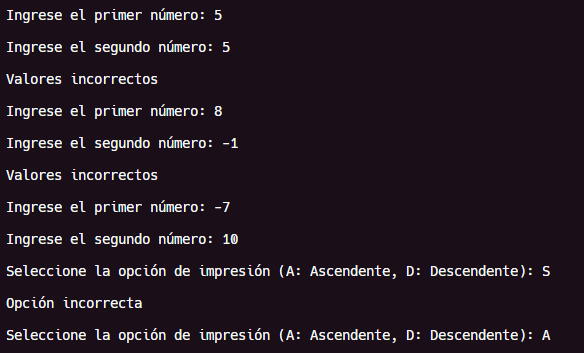
* + La función rand que permite generar números aleatorios es: rand()%(HASTA- DESDE+1)+DESDE;



## 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:



## PRESENTACIÓN

Al finalizar tu laboratorio deberás subir:

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