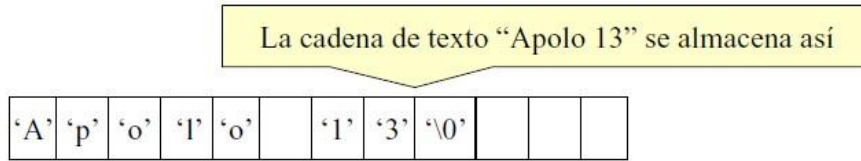


Cadenas de Caracteres

Una cadena en C++ es un arreglo unidimensional de caracteres, terminados con el carácter nulo.



Una cadena se declara de la siguiente manera:

char identificador[tamaño_maximo + 1];

Cuando se declara una cadena hay que tener en cuenta que tendremos que reservar una posición para almacenar el carácter nulo terminador, de modo que si queremos almacenar la cadena "CASA", tendremos que declarar la cadena como:

char valor[5];

Las cuatro primeras posiciones se usan para almacenar los caracteres "CASA" y la posición extra, para el carácter nulo.

En un programa C++, una cadena puede almacenar informaciones en forma de texto, como nombres de personas, mensajes de error, números de teléfono, etc.

- **La asignación directa sólo está permitida cuando se hace junto con la declaración.**

char nombre[]="jose";

o también:

char *nombre="jose";

El siguiente ejemplo producirá un error en el compilador:

char nombre[5];

nombre="jose";

- **La comparación directa de cadenas no está permitida.** Por ejemplo el siguiente ejemplo producirá un error.

if (nombre=="jose")

Para la manipulación de cadenas de caracteres se usa la biblioteca de funciones "string.h"

Funciones de manipulación de cadenas de caracteres

Estas funciones se encuentran en el archivo <string.h>

1) strlen() : devuelve el número de caracteres que hay en una cadena, excluyendo el carácter nulo de terminación de cadena.

Sintaxis: int strlen(cadena)

Ejemplo:

```
char mensaje[]="computación";
```

```
cout<<"La cadena "<<mensaje<<" tiene "<<strlen(mensaje)<<" caracteres "<<endl;
```

2) strcpy() : Esta función permite copiar una cadena en otra
void strcpy(cadena1, cadena2);

Copia el contenido de cadena2, en cadena1.

Ejemplo:

```
char cadena2[]="Computacion";
```

```
char cadena1[80];
```

```
strcpy(cadena1,cadena2);
```

```
cout<<"La cadena 1 es : "<<cadena1<<endl;
```

3) strcmp(): Permite comparar 2 cadenas de caracteres

```
int strcmp(cadena1, cadena2)
```

Esta función devuelve un número entero:

= 0 Si las cadenas son iguales

> 0 Si cadena1 es mayor que cadena2

< 0 Si cadena1 es menor que cadena2

La comparación es lexicográfica.

Ejemplo:

```
char cadena1[80],cadena2[80];
```

```
cout<<"Ingrese primera cadena : ";
```

```
cin>>cadena1;
```

```
cout<<"Ingrese segunda cadena: ";
```

```
cin>>cadena2;
```

```
if(strcmp(cadena1,caden2)==0)
```

```
    cout<<"Las cadenas son iguales "<<endl;
```

```
else
```

```
    if(strcmp(cadena1,cadena2)>0)
```

```
        cout<<"La cadena mayor es : "<<cadena1<<endl;
```

```
else
```

```
    cout<<"La cadena mayor es : "<<cadena2<<endl;
```

4) strcat() : Esta función permite concatenar dos cadenas

```
void strcat(cadena1,cadena2);
```

Une cadena2 al final de cadena1

Ejemplo:

```
char cadena1[]="arriba";  
char cadena2[]="peru";  
strcat(cadena1,cadena2);  
cout<<"cadena resultante : "<<cadena1<<endl;
```

5) strrev() : Invierte la cadena de caracteres

```
strrev(cadena);
```

Ejemplo:

```
char cadena[]="sistemas";  
strrev(cadena);  
cout<<"Cadena al revés : "<<cadena<<endl;
```

6)strupr() : Función que convierte a mayúsculas una cadena de caracteres.

7) strlwr() : Función que convierte a minúsculas una cadena de caracteres.

Ejemplo:

```
char frase[90];  
cout<<"Ingrese frase: ";  
cin.getline(frase,90);  
strupr(frase);  
cout<<"La frase en mayúsculas es : "<<frase<<endl;  
strlwr(frase);  
cout<<"La frase en minúsculas es : "<<frase<<endl
```

Prueba y conversión de caracteres <ctype.h>

Una biblioteca relacionada `#include <ctype.h>` la cual contiene muchas funciones útiles para convertir y probar caracteres individuales.

Las funciones más comunes para revisar caracteres tienen los siguientes prototipos:

`int isalnum(int c)` -- Verdad si c es alfanumérico.

`int isalpha(int c)` -- Verdad si c es una letra. `int isascii(int c)` -- Verdad si c es ASCII.

`int iscntrl(int c)` -- Verdad si c es un caracter de control. `int isdigit(int c)` -- Verdad si c es un dígito decimal.

`int isgraph(int c)` -- Verdad si c es un caracter imprimible, exceptuando el espacio en blanco.

`int islower(int c)` -- Verdad si c es una letra minúscula.

`int isprint(int c)` -- Verdad si c es un caracter imprimible, incluyendo el espacio en blanco. `int`

`ispunct(int c)` -- Verdad si c es un signo de puntuación.

`int isspace(int c)` -- Verdad si c es un espacio

`int isupper(int c)` -- Verdad si c es una letra mayúscula. `int isxdigit(int c)` -- Verdad si c es un dígito hexadecimal.

Las funciones para conversión de caracteres son:

`int toascii(int c)` -- Convierte c a ASCII o un unsigned char de 7 bits, borrando los bits altos.

`int tolower(int c)` -- Convierte la letra c a minúsculas, si es posible.

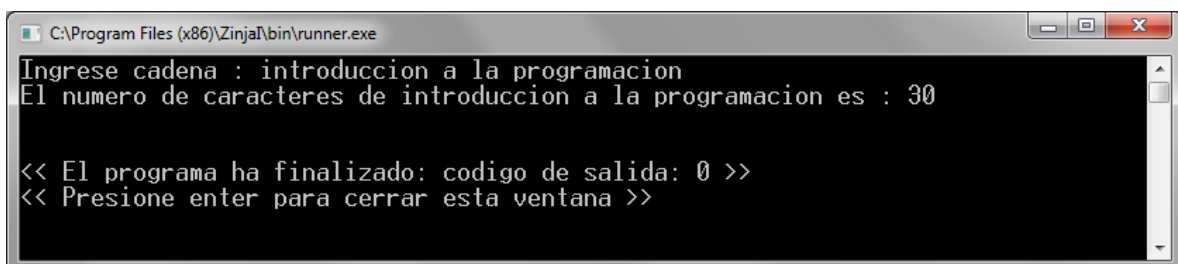
`int toupper(int c)` -- Convierte la letra c a mayúsculas, si es posible.

Ejercicios resueltos

1. Crear una función para calcular el número de caracteres de una cadena.

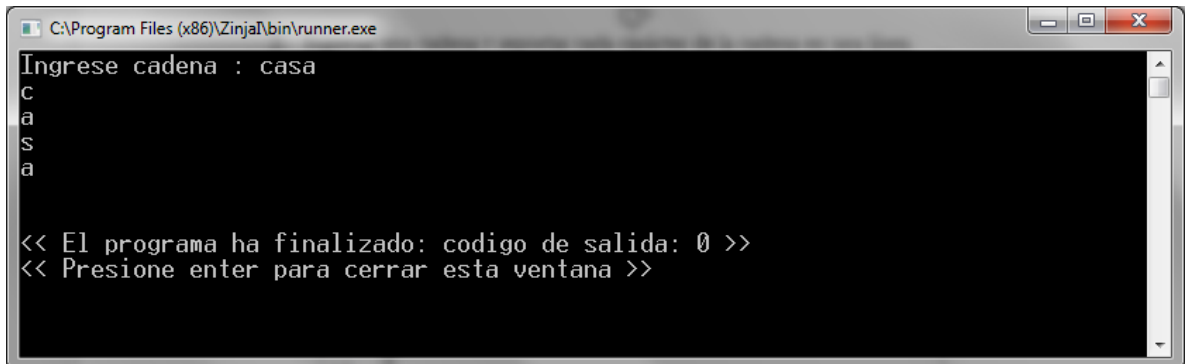
```

1  #include<iostream>
2
3  using namespace std;
4
5  int numCaracteres(char []);
6  int main()
7  {
8      char cadena[80];
9      cout<<"Ingrese cadena : ";
10     cin.getline(cadena,80);
11     cout<<"El numero de caracteres de "<<cadena<<" es : "<<numCaracteres(cadena)<<endl;
12     return 0;
13 }
14
15 int numCaracteres(char cad[])
16 {
17     int i=0; while(cad[i]!=0) i++;
18     return i;
19 }
```



2. Ingresar una cadena y reportar cada carácter de la cadena en una línea.

```
1  #include<iostream>
2  using namespace std;
3  void reportarCaracterPorLinea(char []);
4
5  int main()
6  {
7      char cadena[100];
8      cout<<"Ingrese cadena : ";
9      cin.getline(cadena,100);
10     reportarCaracterPorLinea(cadena);
11     return 0;
12 }
13
14
15 void reportarCaracterPorLinea(char cad[])
16 {
17     int i=0;
18     for(i=0;cad[i]!=0;i++)
19         cout<<cad[i]<<endl;
20 }
21
```

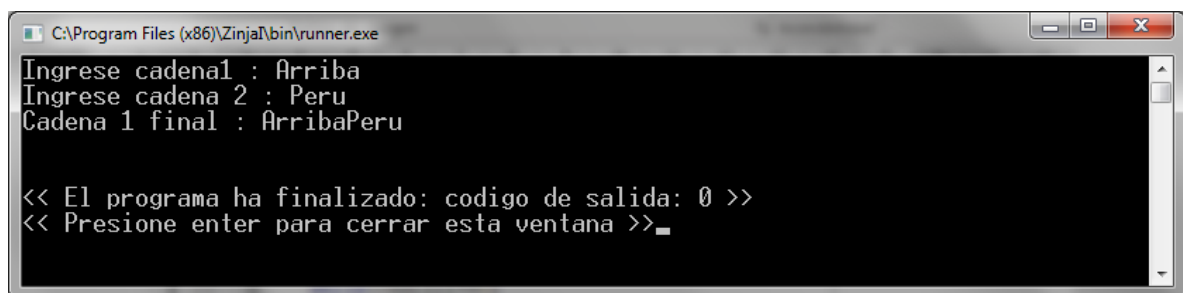


```
C:\Program Files (x86)\Zinja\bin\runner.exe
Ingrese cadena : casa
c
a
s
a

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>
```

3. Escribir una función para concatenar 2 cadenas de caracteres. Si tenemos dos cadenas cad1 y cad2 colocar todos los caracteres de cad2 al final de cad1. Si cad1="Arriba" y cad2="Peru" y concatenamos cad1 con cad2, al final cad1 es "ArribaPeru".

```
1  #include<iostream>
2
3  using namespace std;
4
5  void concatenar(char [], char []);
6
7  int main()
8  {
9      char cad1[100], cad2[100];
10     cout<<"Ingrese cadena1 : ";
11     cin.getline(cad1,100);
12     cout<<"Ingrese cadena 2 : ";
13     cin.getline(cad2,100);
14     concatenar(cad1,cad2);
15     cout<<"Cadena 1 final : "<<cad1<<endl;
16     return 0;
17 }
18
19 void concatenar(char cad1[], char cad2[])
20 {
21     int i=0,j=0;
22     while(cad1[i]!=0)
23         i++;
24
25     while(cad2[j]!=0)
26     {
27         cad1[i]=cad2[j];
28         i++;
29         j++;
30     }
31     cad1[i]=0;
32 }
```

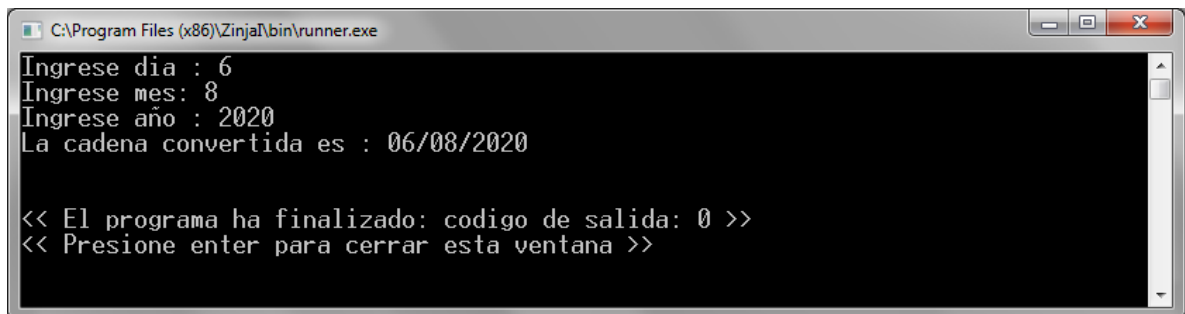


```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Ingrese cadena1 : Arriba
Ingrese cadena 2 : Peru
Cadena 1 final : ArribaPeru

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>_
```

4. Ingresar una fecha en los enteros día, mes y año por día: 7 mes:10 año: 2011 y obtener una cadena por ejemplo: "07/10/2011"

```
1  #include<iostream>
2  using namespace std;
3  void convertir(int , int , int ,char []);
4
5  int main()
6  {
7      int d, m, a;
8      char cadena[20];
9      cout<<"Ingrese dia : ";
10     cin>>d;
11     cout<<"Ingrese mes: ";
12     cin>>m;
13     cout<<"Ingrese año : ";
14     cin>>a;
15     convertir(d,m,a,cadena);
16     cout<<"La cadena convertida es : "<<cadena<<endl;
17     return 0;
18 }
19
20 void convertir(int d, int m, int a,char cad[])
21 {
22     cad[0] = 48 + d/10;
23     cad[1] = 48 + d%10;
24     cad[2]= '/';
25     cad[3] = 48 + m/10;
26     cad[4] = 48 + m%10;
27     cad[5]= '/';
28     cad[6]= 48 + a/1000;
29     a=a%1000;
30     cad[7]= 48 + a/100;
31     a=a%100;
32     cad[8]= 48 + a/10;
33     cad[9]= 48 + a%10;
34     cad[10]=0;
35 }
```



```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Ingrese dia : 6
Ingrese mes: 8
Ingrese año : 2020
La cadena convertida es : 06/08/2020

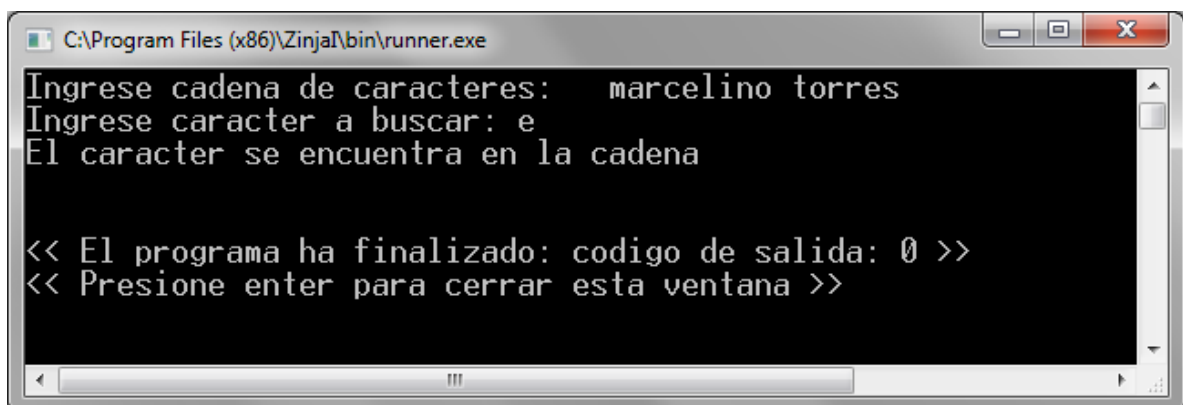
<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>
```

5. Ingresar una cadena que representa la hora como por ejemplo: 12:20:35 y obtener en variables enteras la hora, minutos y segundos, h=12, m=20 y s=35.

```
1  #include<iostream>
2
3  using namespace std;
4
5  void obtenerEnteros(char [], int &,int &,int &);
6  void reporte(int,int,int);
7
8  int main()
9  {
10     char hora[100];
11     int h,m,s;
12     cout<<"Ingrese (hh:mm:ss) : ";
13     cin>>hora;
14     obtenerEnteros(hora,h,m,s);
15     reporte(h,m,s);
16     return 0;
17 }
18
19 void obtenerEnteros(char hora[], int &h, int &m,int &s)
20 {
21     h=(hora[0]-48)*10+(hora[1]-48);
22     m=(hora[3]-48)*10+(hora[4]-48);
23     s=(hora[6]-48)*10+(hora[7]-48);
24 }
25
26 void reporte(int h, int m, int s)
27 {
28     cout<<"Horas : "<<h<<endl;
29     cout<<"Minutos: "<<m<<endl;
30     cout<<"Segundos : "<<s<<endl;
31 }
```

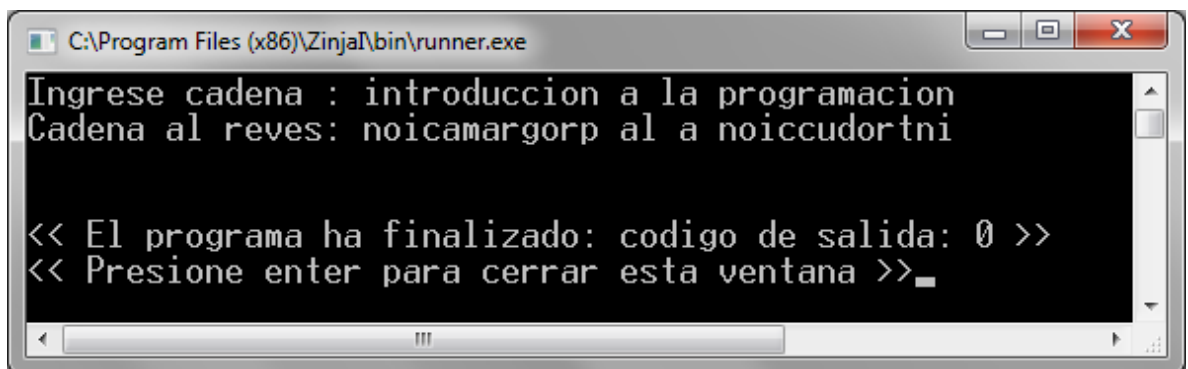

6. Escribir una función `bool busca_caracter(char cad[], char c)` que busque un carácter `c` en una cadena `cad`, devolviendo un valor lógico Cierto si lo encuentra y Falso en caso contrario.

```
1  #include<iostream>
2
3  using namespace std;
4  bool busca_caracter(char cad[], char c);
5
6  int main()
7  {
8      char cad[100],car;
9      cout<<"Ingrese cadena de caracteres:  ";
10     cin.getline(cad,100);
11     cout<<"Ingrese caracter a buscar: ";
12     cin>>car;
13     if(busca_caracter(cad,car))
14         cout<<"El caracter se encuentra en la cadena"<<endl;
15     else
16         cout<<"El caracter no se encuentra en la cadena"<<endl;
17     return 0;
18 }
19
20 bool busca_caracter(char cad[], char c)
21 {
22     int i;
23     for(i=0;cad[i]!=0;i++)
24     {
25         if(cad[i]==c)
26             return true;
27     }
28     return false;
29 }
```



7. Escribir una función para invertir una cadena de caracteres.

```
1  #include<iostream>
2  #include<string.h>
3
4  using namespace std;
5
6  void invertirCadena(char []);
7
8  int main()
9  {
10     char cadena[100];
11     cout<<"Ingrese cadena : ";
12     cin.getline(cadena,100);
13     invertirCadena(cadena);
14     cout<<"Cadena al revés: "<<cadena<<endl;
15     return 0;
16 }
17
18 void invertirCadena(char cad[])
19 {
20     int n,i,j;
21     char temp;
22     n=strlen(cad);
23     for(i=0,j=n-1;i<n/2;i++,j--)
24     {
25         temp=cad[i];
26         cad[i]=cad[j];
27         cad[j]=temp;
28     }
29 }
```

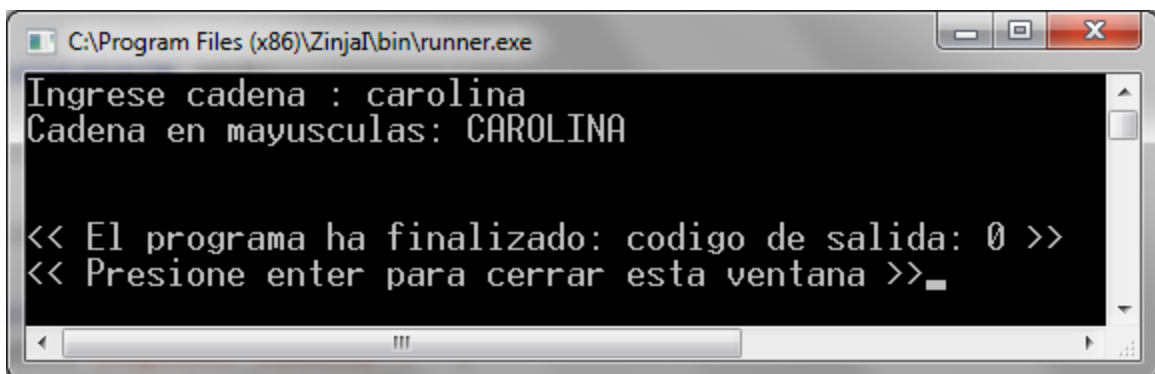


```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Ingrese cadena : introduccion a la programacion
Cadena al revés: noicamargorp al a noiccudortni

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>_
```

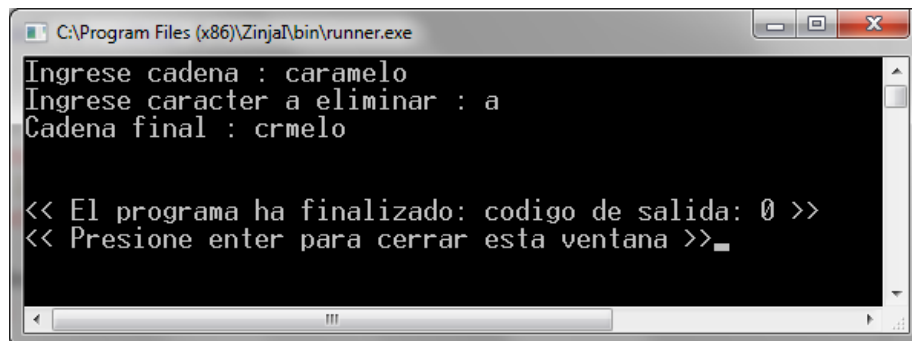
8. Escribir una función para convertir una cadena de minúsculas a mayúsculas.

```
1  #include<iostream>
2
3  using namespace std;
4
5  void convertirAMayusculas(char []);
6
7  int main()
8  {
9      char cadena[100];
10     cout<<"Ingrese cadena : ";
11     cin.getline(cadena,100);
12     convertirAMayusculas(cadena);
13     cout<<"Cadena en mayusculas: "<<cadena<<endl;
14     return 0;
15 }
16 void convertirAMayusculas(char cad[])
17 {
18     int i;
19     for(i=0;cad[i]!=0;i++)
20     {
21         if(cad[i]>='a' && cad[i]<='z')
22             cad[i]=cad[i]-32;
23     }
24 }
```



9. Hacer un programa para ingresar una cadena y luego ingresar un carácter y eliminar el carácter todas las veces que se encuentre en la cadena.

```
1  #include<iostream>
2  using namespace std;
3  void eliminaCaracter(char [], char );
4  int main()
5  {
6      char cadena[100],car;
7      cout<<"Ingrese cadena : ";
8      cin.getline(cadena,100);
9      cout<<"Ingrese caracter a eliminar : ";
10     cin>>car;
11     eliminaCaracter(cadena,car);
12     cout<<"Cadena final : "<<cadena<<endl;
13     return 0;
14 }
15
16 void eliminaCaracter(char cad[],char car)
17 {
18     int i,j,n;
19     n=strlen(cad);
20     for(i=0;i<n;i++)
21     {
22         if(cad[i]==car)
23         {
24             for(j=i;j<n;j++)
25                 cad[j]=cad[j+1];
26             n=n-1;
27             i--;
28         }
29     }
30 }
```



```
C:\Program Files (x86)\Zinja\bin\runner.exe
Ingrese cadena : caramelo
Ingrese caracter a eliminar : a
Cadena final : crmelo

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>
```