
SISTEMAS OPERATIVOS I - Práctica 2 (Parte 1)
Grado en Ingeniería Informática - Escuela Superior de Informática (UCLM)

1. Actividades de Laboratorio

Escriba un programa C estándar para cada uno de los distintos enunciados con la funcionalidad indicada. Salvo que se especifique lo contrario, se entenderá que la entrada y salida del programa corresponderá a la entrada estándar y salida estándar.

1. La salida del programa será una tabla de conversión de grados Fahrenheit a grados Celsius comenzando desde 0º grados Fahrenheit a 300º grados Fahrenheit en incrementos de 20º

```
1  #include <stdio.h>

3  int main() {
    int fahr;

5      for (fahr = 0; fahr <= 300; fahr = fahr + 20)
7          printf("%3d %6.1f\n", fahr, (5.0 / 9.0) * (fahr - 32));

9      return 0;
    }
```

2. La salida del programa será una copia de la entrada. Es decir, la funcionalidad del comando **cat**

```
    #include <stdio.h>

2
    int main() {
4        int c;

6        while ((c = getchar()) != EOF)
            putchar(c);

8        return 0;
10    }
```

3. La salida del programa será la cuenta de caracteres de la entrada. Es decir, la funcionalidad del comando **wc -c**

```
    #include <stdio.h>

2
    int main() {
4        int nc;

6        for (nc = 0; getchar() != EOF; ++nc);

8        printf("%d\n", nc);

10       return 0;
    }
```

4. La salida del programa será la cuenta de líneas de la entrada. Es decir, la funcionalidad del comando **wc -l**

```
1  #include <stdio.h>

3  int main() {
    int c, nl;

5      nl = 0;

    }
```

```

7      while ((c = getchar()) != EOF)
          if (c == '\n') /* La cuenta de líneas es la cuenta de '\n' */
10         ++nl;
11
12     printf("%d\n", nl);
13
14     return 0;
15 }

```

5. La salida del programa será la cuenta de palabras de la entrada. Es decir, la funcionalidad del comando `wc -w`. Se considerarán como separadores de palabras los caracteres blancos, los tabuladores horizontales y los caracteres de nueva línea

```

#include <stdio.h>
2
#define IN 1 /* En una palabra */
4 #define OUT 0 /* Fuera de una palabra */

6 int main() {
    int c, nw, state;
8
    state = OUT;
10    nw = 0;

12    while ((c = getchar()) != EOF) {
        if (c == ' ' || c == '\n' || c == '\t')
14            state = OUT;
        else if (state == OUT) {
16            state = IN;
            ++nw;
18        }
    }
20
    printf("%d\n", nw);
22
    return 0;
24 }

```

6. La salida del programa será la cuenta de palabras de la entrada que empiecen por un carácter alfabético. Se considerarán como separadores de palabras los caracteres blancos, los tabuladores horizontales y los caracteres de nueva línea

```

#include <stdio.h>
2 #include <ctype.h>

4 #define IN 1 /* En una palabra */
#define OUT 0 /* Fuera de una palabra */
6

8 int main() {
    int c, nw, state;

10    state = OUT;
    nw = 0;

12    while ((c = getchar()) != EOF) {
14        if (c == ' ' || c == '\n' || c == '\t')
            state = OUT;
        else if (state == OUT) {
16            state = IN;
            if (isalpha(c))
18                ++nw;
20        }
    }

```

```

    }
22     printf("%d\n", nw);
24
    return 0;
26 }
```

7. La salida del programa será la cuenta de caracteres, líneas y palabras de la entrada. Es decir, la funcionalidad del comando `wc -clw`. Se considerarán como separadores de palabras los caracteres blancos, los tabuladores horizontales y los caracteres de nueva línea

```

#include <stdio.h>
2
#define IN 1 /* En una palabra */
4 #define OUT 0 /* Fuera de una palabra */

6 int main() {
    int c, nl, nw, nc, state;
8
    state = OUT;
10    nl = nw = nc = 0;

12    while ((c = getchar()) != EOF) {
        ++nc;
14
        if (c == '\n')
16            ++nl;

18        if (c == ' ' || c == '\n' || c == '\t')
            state = OUT;
20        else if (state == OUT) {
            state = IN;
22            ++nw;
        }
24    }

26    printf("%d %d %d\n", nl, nw, nc);

28    return 0;
}
```

8. Copiar la entrada a la salida reemplazando cada cadena de uno o más blancos por un sólo blanco.

```

1  #include <stdio.h>
   #include <ctype.h>
3
   int main() {
7       int c;

       while ((c = getchar()) != EOF) {
           putchar(c);
9
           if (c == ' ') {
11              while ((c = getchar()) == ' ') ; /* Cuerpo vacio del while */
               if (c != EOF) putchar(c);
13          }
       }
15
       return 0;
17 }
```

9. La salida del programa será la cuenta de cada dígito, los separadores agrupados en una única clase y otra cuenta con la clase del resto de caracteres. Se considerarán como separadores de palabras los caracteres blancos, los tabuladores horizontales y los caracteres de nueva línea

```

1  #include <stdio.h>

3  int main() {
    int i, c;
5   int ndigitos[10], nblancos, notros;

7   for (i = 0; i < 10; i++) /* Inicialización de la cuenta de dígitos */
        ndigitos[i] = 0;

9   nblancos = notros = 0;

11  while ((c = getchar()) != EOF)
13      if (c >= '0' && c <= '9') /* Dígitos */
          ndigitos[c - '0']++;
15      else
          if (c == ' ' || c == '\t' || c == '\n') /* Blancos */
17          nblancos++;
          else
19          notros++; /* Otros */

21  printf("Dígitos: ");
  for (i = 0; i < 10; i++)
23      printf("%d ", ndigitos[i]);
  printf("Blancos: %d Otros: %d\n", nblancos, notros);

25

27  return 0;
  }

```

10. El programa debe determinar cuál es el carácter más frecuente en la entrada y el número de veces que aparece. La salida del programa debe ser una línea con el siguiente formato: <carácter_más_frecuente>:<frecuencia>

```

1  #include <stdio.h>

3  #define N_ASCII 256

5  int main() {
    int i, c;
7   int c_freq; /* Carácter más frecuente */
    int n_freq; /* Frecuencia del carácter más frecuente */
9   int ncaracteres[N_ASCII]; /* Tabla de frecuencias de todos los caracteres */

11  for (i = 0; i < N_ASCII; i++) /* Inicialización de la tabla de frecuencias */
        ncaracteres[i] = 0;

13

15  while ((c = getchar()) != EOF) /* Obtención de las frecuencias */
        ncaracteres[c]++;

17  /* Obtención del carácter más frecuente */
  for (i = 0, n_freq = -1, c_freq = -1; i < N_ASCII; i++)
19      if (ncaracteres[i] > n_freq) {
          n_freq = ncaracteres[i];
21          c_freq = i;
      }

23

25  if (n_freq > 0) printf ("%c:%d\n", c_freq, n_freq);

27  return 0;
  }

```

11. Escriba una función que cuente los bits con valor a 1 de su parámetro de entrada del tipo entero sin signo y un programa que utilice dicha función

```
1  #include <stdio.h>

3  int bitcount(unsigned int x);

5  int main() {

7      printf("%d\n", bitcount(7));

9      return 0;
10 }

11 int bitcount(unsigned int x) {
12     int b;

13     for (b = 0; x != 0; x >>= 1)
14         if (x & 01)
15             b++;

16     return b;
17 }
```