

Realizar un programa que cuente los dígitos de un fichero introducido y los muestre por pantalla.

```
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>

#define DIGITOS 10

void contardigitos(FILE *fp, int *digitos);

int main (int argc, char *argv[]){
    int i;
    FILE *fp;
    int digitos[DIGITOS];
    for(i=0; i<DIGITOS; i++){
        digitos[i]=0;
    }

    if(--argc<1){
        fprintf(stderr, "Error, no se ha introducido fichero\n");
        return 1;
    }
    else{
        while(argc-->0){
            if((fp=fopen(*++argv, "r"))==NULL){
                fprintf(stderr, "Error. no es posible abrir %s\n", *argv);
                return 1;
            }
            contardigitos(fp, digitos);
        }
    }

    for(i=0; i<DIGITOS; i++){
        printf("%d --> %d\n", i, digitos[i]);
    }

    return 0;
}

void contardigitos(FILE *fp, int *digitos){
    int c;
    while ((c=fgetc(fp))!=EOF){
        if(c>='0' && c<='9'){
            digitos[c-'0']++;
        }
    }
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

#define DIGITOS 10

void filecopy(FILE *ifp, int *digitos);

int main (int argc, char *argv[]){
    int digitos[DIGITOS];
    int i;
    FILE *fp;

    for(i=0; i<DIGITOS; i++){
        digitos[i]=0;
    }

    if(--argc < 1){
        filecopy(stdin, digitos);
    }
    else {
        while (argc-->0){
            if((fp=fopen(*++argv, "r"))==NULL){
                fprintf(stderr, "Error, no es posible abrir el fichero %s\n", *argv);
                return EXIT_FAILURE;
            }
            filecopy(fp, digitos);
        }
    }

    for(i=0; i<DIGITOS; i++){
        printf("%d-->%d\n", i, digitos[i]);
    }

    return EXIT_SUCCESS;
}

void filecopy(FILE *ifp, int *digitos){
    int c;
    while ((c=fgetc(ifp))!=EOF){
        if(c>='0' && c<='9'){
            digitos[c-'0']++;
        }
    }
}
```

Realizar un programa que cuente los caracteres de un fichero y si no se introduce fichero, que cuente los de la entrada estándar.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#define DIGITOS 10

int main(int argc, char *argv[]) {
    int c, i;
    int digitos[DIGITOS];
    int otros=0;
    int opcion_v=0;
    int opcion_U=0;
    int opcion_o=0;
    FILE *fp;

    for(i=0; i<DIGITOS; i++){
        digitos[i]=0;
    }

    while (--argc > 0 && (argv[0]!='-')) {
        while ((c=argv[0])) {
            switch(c) {
                case 'v':
                    if(opcion_U){
                        fprintf(stderr, "Error, opciones -v y -U incompatibles\n");
                        argc=0;
                        return EXIT_FAILURE;
                    } else {
                        opcion_v=1;
                    }
                    break;
                case 'U':
                    if(opcion_v){
                        fprintf(stderr, "Error, opciones -v y -U incompatibles\n");
                        argc=0;
                        return EXIT_FAILURE;
                    } else {
                        opcion_U=1;
                    }
                    break;
                case 'o':
                    opcion_o=1;
                    break;
                default:
                    fprintf(stderr, "Error, opcion no valida -%s\n", argv[0]);
                    argc=0;
                    return EXIT_FAILURE;
                    break;
            }
        }
    }

    if (argc < 1) {
        fprintf(stderr, "Error, no ha introducido fichero\n");
        return EXIT_FAILURE;
    }
    else {
        while(argc-->0) {
            if((fp=fopen(argv++, "r"))==NULL) {
                fprintf(stderr, "Error al abrir el fichero %s\n", argv);
                return EXIT_FAILURE;
            }
            while((c=fgetc(fp))!=EOF) {
                if (c>='0' && c<='9') {
                    digitos[c-'0']++;
                } else {
                    otros++;
                }
            }
            fclose(fp);
        }

        if(opcion_v) {
            for(i=0; i<DIGITOS; i++) {
                printf("%d:%d/", i, digitos[i]);
            }
        }
        if(opcion_U) {
            for(i=0; i<DIGITOS; i++) {
                printf("%d-->%d\n", i, digitos[i]);
            }
        }
        if(opcion_o) {
            printf("Otros: %d", otros);
        }
        printf("\n");
        return EXIT_SUCCESS;
    }
}

```

Crear un programa llamado **cuentadigitos** que imprima en la salida estándar el número de dígitos que contiene el fichero cuyo nombre se pasa como argumento en la línea de órdenes. El programa admitirá las siguientes opciones:

- v el programa imprimirá el resultado de la manera 0:X / 1:X /2: X..
- V el programa imprimirá el resultado de la manera 0→X\n 1→X\n
- ...

- o el programa imprimirá la cuenta de otros caracteres Otros: X\n

Donde X es el numero en que se repite dicho valor. La función -v y -V son incompatibles.

La sintaxis en la línea de ordenes será:

Cuentadigitos [-o] [-v] [-V] [<archivo>]+

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

int main(int argc, char *argv[]) {
    int c;
    int opcion_a=0;
    int opcion_A=0;
    int opcion_M=0;
    int opcion_m=0;
    FILE *fp;
    while(--argc > 0 && (**argv)[0]!='-'){
        while((c=**argv[0])){
            switch(c){
                case 'a':
                    if(opcion_A){
                        fprintf(stderr, "Error. opcion -a y -A incompatibles\n");
                        argc=0;
                        return EXIT_FAILURE;
                    } else {
                        opcion_a=1;
                    }
                    break;
                case 'A':
                    if(opcion_a){
                        fprintf(stderr, "Error.Opcion -a y -A incompatibles\n");
                        argc=0;
                        return EXIT_FAILURE;
                    } else {
                        opcion_A=1;
                    }
                    break;
                case 'm':
                    if(opcion_M){
                        fprintf(stderr, "Error.Opciones -m y -M incompatibles\n");
                        argc=0;
                        return EXIT_FAILURE;
                    } else {
                        opcion_m=1;
                    }
                    break;
                case 'M':
                    if(opcion_m){
                        fprintf(stderr, "Error.Opciones -m y -M incompatibles\n");
                        argc=0;
                        return EXIT_FAILURE;
                    } else {
                        opcion_M=1;
                    }
                    break;
                default:
                    fprintf(stderr, "Error. Opcion ilegal -%s\n", *argv);
                    argc=0;
                    return EXIT_FAILURE;
                    break;
            }
        }
    }

    if(argc < 1){
        fprintf(stderr, "Error.Sintaxis sustituir [-m] [-M] [-a] [-A] [<archivo>]+\n");
        return EXIT_FAILURE;
    } else {
        while (argc-- > 0){
            if ((fp=fopen(*argv++, "r"))==NULL){
                fprintf(stderr, "Error al abrir el fichero %s\n", *argv);
                return EXIT_FAILURE;
            }
            while ((c=fgetc(fp))!=EOF){
                if( c>='0' && c<='9'){
                    if(opcion_a){
                        putchar('0');
                    }
                    else if(opcion_A){
                        putchar('#');
                    }
                    else{
                        putchar(c);
                    }
                } else {
                    if(opcion_m){
                        c=tolower(c);
                    }
                    if(opcion_M){
                        c=toupper(c);
                    }
                    putchar(c);
                }
            }
            fclose(fp);
        }
    }

    return EXIT_SUCCESS;
}

```

Crear un programa llamado **sustituir** que imprima en la salida estándar el contenido del fichero según las siguientes opciones:

-a El programa sustituirá los dígitos por el carácter '@'

-A El programa sustituirá los dígitos por el carácter '#'

-M El programa mostrara todo en mayúsculas

-m el programa mostrara todo en minúsculas

Las opciones -a y -A son incompatibles, al igual que las opciones -m y -M

Si no se pone ninguna opción, se imprimirá el fichero tal cual.

La sintaxis en la línea de ordenes será:

Sustituir [-m] [-M] [-a] [-A] [<archivo>]+

Realizar un programa que implemente la función del comando cat

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

void filecopy(FILE *ifp, FILE *ofp);

int main (int argc, char *argv[]) {
    FILE *fp;

    if (--argc == 0) {
        filecopy(stdin, stdout);
    }
    else {
        while ((argc-- > 0)) {
            if ((fp = fopen(++argv, "r")) == NULL) {
                fprintf(stderr, "Error, no se puede abrir el fichero %s\n", *argv);
                return EXIT_FAILURE;
            }
            filecopy(fp, stdout);
            fclose(fp);
        }
    }
    return EXIT_SUCCESS;
}

void filecopy(FILE *ifp, FILE *ofp) {
    int c;
    while ((c = getc(ifp)) != EOF) {
        putc(c, ofp);
    }
}
```

Realizar un programa que imite el comando HEAD, el programa recibirá una opción -x donde x es el número de líneas a mostrar, si no se indica nada mostrara las 3 primeras líneas.

Headc [-<n>] [<fichero>]+

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

void mostrarlinea(FILE *fp, int linea);

int main(int argc, char *argv[]) {
    int linea=3;
    int c;
    FILE *fp;

    while (--argc > 0 && (*argv)[0] != '-') {
        while ((c = **argv)) {
            if (isdigit(c)) {
                linea = atoi(argv[0]);
                argv[0] = argv[0] + strlen(argv[0]) - 1;
            }
            if (c < 0) {
                fprintf(stderr, "Error, numero invalido");
                return EXIT_FAILURE;
            }
        }
        else {
            fprintf(stderr, "Error, argumento ilegal");
            return EXIT_FAILURE;
        }
    }

    if (argc < 1) {
        fprintf(stderr, "Error, no ha introducido fichero");
        return EXIT_FAILURE;
    }
    else {
        while (argc-- > 0) {
            if ((fp = fopen(*argv++, "r")) == NULL) {
                fprintf(stderr, "Error, no es posible abrir el dichero");
                return EXIT_FAILURE;
            }
            mostrarlinea(fp, linea);
            fclose(fp);
        }
    }
    return EXIT_SUCCESS;
}

void mostrarlinea(FILE *fp, int linea) {
    int c, i;
    i = 0;

    while ((c = fgetc(fp)) != EOF && i < linea) {
        putchar(c);
        if (c == '\n') {
            i++;
        }
    }
}
```

Realizar un programa que reciva un valor -x donde X es un numero y trunque el contenido del fichero por lineas su contenido

truncar [-<n>] [<archivo>]+

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

void truncar (FILE *ifp, int trunc);

int main(int argc, char *argv[]){
    int c;
    int trunc;
    int opcion=0;
    FILE *fp;

    while (--argc && (++argv)[0]!='-'){
        while((c=**++argv[0])){
            if(isdigit(c)){
                if(opcion){
                    fprintf(stderr,"Error. Opcion ilegal\n");
                    return EXIT_FAILURE;
                }
                trunc=atoi(argv[0]);
                argv[0]=argv[0]+strlen(argv[0])-1;
                if(trunc<0){
                    fprintf(stderr,"Error. Opcion ilegal\n");
                    return EXIT_FAILURE;
                }
                opcion=1;
            }
            else {
                fprintf(stderr,"Error,Opcion ilegal -%s\n",*argv);
                return EXIT_FAILURE;
            }
        }
        if(!opcion){
            fprintf(stderr,"Error.No ha seleccionado Opcion '\n");
            return EXIT_FAILURE;
        }
        if (argc < 1){
            fprintf(stderr,"Error. No ha introducido fichero \n");
            return EXIT_FAILURE;
        }
        else{
            while(argc-- > 0){
                if((fp=fopen(*argv++, "r"))==NULL){
                    fprintf(stderr,"Error al abrir el fichero %s\n",*argv);
                    return EXIT_FAILURE;
                }
                truncar(fp, trunc);
            }
            fclose(fp);
        }
    }

    return EXIT_SUCCESS;
}

void truncar (FILE *ifp, int trunc){
    int c,i;
    i=0;
    while((c=fgetc(ifp))!=EOF){
        if(i<trunc){
            putchar(c);
            i++;
        }
        if(c=='\n'){
            putchar(c);
            i=0;
        }
    }
}
```