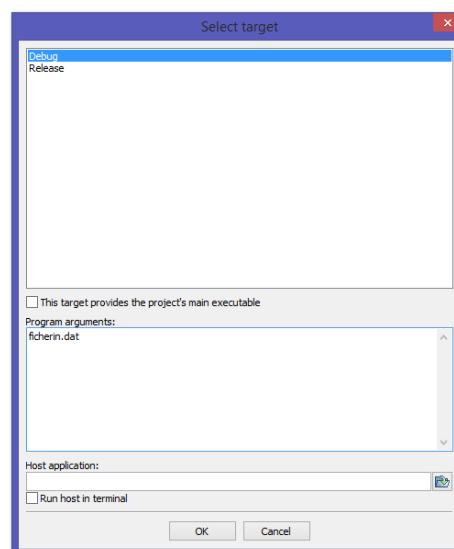


Esta práctica es la continuación de la práctica anterior y se deben realizar las siguientes acciones:

1. Acabar la práctica de la semana pasada si no se hubiera terminado.
2. Una vez que se ha ejecutado correctamente y se ha obtenido el fichero de texto con las apuestas premiadas, repartir la práctica en tres ficheros de compilación por separado. Para ello se deberá añadir a cada uno de los dos proyectos un fichero **tipos.h** y otro **funciones.c** en los que se va a repartir el código de esta manera (explicada en las transparencias que hay en moodle):
  - a. en el .h se pondrán las definiciones de constantes, tipos y los prototipos de las funciones implementadas, haciendo uso de la directiva de compilación **#ifndef .....#endif** como se explica en las transparencias.
  - b. en el .c la implementación de las funciones, incluyendo los ficheros `stdio.h`, `stdlib.h` y `time.h` como estaban en el fichero del main.
  - c. el main se quedará solo en el fichero .c en el que está.
  - d. se incluirá el fichero .h en los dos .c usando " " en vez de < > para ello.
  - e. se compilarán y ejecutarán los dos programas para comprobar que lo hacen correctamente.
3. Hacer los cambios necesarios en los dos programas para que reciban los nombres de los ficheros como parámetros de main (según se ha explicado en clase y recogido en las transparencias de moodle) en vez de leerlos por teclado.

Una vez hechos los cambios y compilado correctamente, se deberá establecer en **Code::blocks** los parámetros para cada uno de los programas. Para ello se elije la opción **Project/Set program's arguments...** y aparecerá la siguiente pantalla:



y en la ventana Program arguments se pone el nombre del fichero a generar en el primer program y, en el segundo, los nombres del fichero de apuestas que se haya generado con el primer programa y el de texto que se va a generar, uno en cada línea.

Una vez establecidos los argumentos, ejecutar los programas en el orden adecuado y comprobar que todo funciona correctamente. Habrá que copiar el fichero generado en el primer programa en la carpeta en la que está el segundo, como se hacía en la práctica de la semana pasada.

4. En la carpeta `bin\debug` de cada uno de los proyectos están los ficheros ejecutables de cada uno de ellos (.exe). Crear una carpeta nueva y copiar en ella estos dos ficheros, de forma que sean los únicos ficheros en la carpeta. A continuación ejecutar desde Windows (pulsar en el icono de Windows de la barra de tareas con el botón derecho del ratón y elegir la opción **Ejecutar**, poner el nombre del primer programa y el nombre del fichero que se quiera generar y dar a Intro. Si es preciso, dar el camino completo hasta el fichero a ejecutar).  
Comprobar que se ha generado el fichero de datos en la carpeta.  
Hacer lo mismo con el segundo programa, pasándole los nombres de los dos ficheros como parámetros.  
Comprobar que se ha generado el fichero de texto en la carpeta.

Solución programa 1:

Fichero .h

```
#ifndef TIPOS
#define TIPOS
#include <stdio.h>

typedef struct
{
    char id[11];
    unsigned n[6];
} Apuesta;

int random(int, int);
void crear_id(char [11]);
void crear_num(unsigned [6]);
void crear_apuesta(Apuesta*);
void rellenar_fapuestas(FILE* f_in, unsigned);
unsigned acertados(Apuesta , unsigned[6], unsigned *);

#endif // TIPOS
```

Fichero .c (de funciones)

```
#include "tipos.h"
#include <stdio.h>
#include <stdlib.h>

void rellenar_fapuestas(FILE* f_in, unsigned _ap)
{
    int i;
    Apuesta ap;
    for (i=0; i<_ap; i++)
    {
        crear_apuesta(&ap);
        fwrite(&ap, sizeof(Apuesta), 1, f_in);
    }
}

void crear_id(char _id[11])
{
    int i;
    _id[0] = '2';
    _id[1] = '8';
    for (i=2; i<10; i++)
        _id[i] = random('0', '9');
    _id[10] = random('A', 'Z');
    _id[11] = '\0';
}

void crear_num(unsigned _num[6])
{
    int i, j;
    for (i=0; i<6;)
    {
        _num[i] = random(1, 49);
        for (j=0; j<i && _num[j] != _num[i]; j++);
        if (j == i)
            i++;
    }
}

void crear_apuesta(Apuesta* _ap)
{
    crear_id(_ap->id);
    crear_num(_ap->n);
}

int random(int a, int b)
{
    return (rand()%(b-a+1)+a);
}
```

```
unsigned acertados(Apuesta _ap, unsigned _premiado[6], unsigned * _acertados)
{
    unsigned acertados = 0, i, j;
    for (i=0; i<6; i++)
    {
        for(j=0; j<6 && _ap.n[i] != _premiado[j]; j++);
        if (j<6)
            _acertados[acertados++] = _ap.n[i];
    }
    return acertados;
}
```

Fichero .c (del main)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "tipos.h"

int main(int argc, char* argv[])
{
    FILE* f_in;
    unsigned n_ap;

    srand(time(NULL));

    if (argc<2)
    {
        printf("ERROR");
        return 1;
    }

    if (!(f_in=fopen(argv[1], "wb")))
        return 2;

    printf("\nCuántas apuestas quieres generar? :");
    scanf("%u", &n_ap);
    fflush(stdin);

    printf("\nEl tamaño del archivo va a ser %.9fmb Seguir? (S/N): ", (sizeof(Apuesta)/1000000.f));
    if (toupper(getchar()) != 'S')
        return 0;

    rellenar_fapuestas(f_in, n_ap);

    fclose(f_in);
    return 0;
}
```

Solución programa 2:

Fichero .h

```
#ifndef TIPOS
#define TIPOS
#include <stdio.h>

typedef struct
{
    char id[11];
    unsigned n[6];
} Apuesta;

int random(int, int);
void crear_id(char [11]);
void crear_num(unsigned [6]);
void crear_apuesta(Apuesta*);
void rellenar_fapuestas(FILE* f_in, unsigned);
unsigned acertados(Apuesta , unsigned[6], unsigned *);

#endif // TIPOS
```

Fichero .c (de funciones)

```
#include "tipos.h"
#include <stdio.h>
#include <stdlib.h>

void rellenar_fapuestas(FILE* f_in, unsigned _ap)
{
    int i;
    Apuesta ap;
    for (i=0; i<_ap; i++)
    {
        crear_apuesta(&ap);
        fwrite(&ap, sizeof(Apuesta), 1, f_in);
    }
}

void crear_id(char _id[11])
{
    int i;
    _id[0] = '2';
    _id[1] = '8';
    for (i=2; i<10; i++)
        _id[i] = random('0', '9');
    _id[10] = random('A', 'Z');
```

```
_id[11] = '\0';
}

void crear_num(unsigned _num[6])
{
    int i, j;
    for (i=0; i<6;)
    {
        _num[i] = random(1, 49);
        for (j=0; j<i && _num[j] != _num[i]; j++);
        if (j == i)
            i++;
    }
}

void crear_apuesta(Apuesta* _ap)
{
    crear_id(_ap->id);
    crear_num(_ap->n);
}

int random(int a, int b)
{
    return (rand()%(b-a+1)+a);
}

unsigned acertados(Apuesta _ap, unsigned _premiado[6], unsigned * _acertados)
{
    unsigned acertados = 0, i, j;
    for (i=0; i<6; i++)
    {
        for(j=0; j<6 && _ap.n[i] != _premiado[j]; j++);
        if (j<6)
            _acertados[acertados++] = _ap.n[i];
    }
    return acertados;
}
```

Fichero .c (del main)

```
#include "tipo
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char* argv[])
{
    Apuesta ap;
    FILE* fapuestas;
    FILE* fpremiados;
```

```
unsigned premiado[6], acc[6];
unsigned i, j;

srand(time(NULL));

if (argc<3)
    return 1;

if (!(fapuestas=fopen(argv[1], "rb")))
    return 2;
if (!(fpremiados=fopen(argv[2], "w")))
    return 2;

crear_num(premiado);
fprintf(fpremiados, "Combinación ganadora: ");
for (i=0; i<6; i++)
    fprintf(fpremiados, "%u ", premiado[i]);
fprintf(fpremiados, "\n");

while (fread(&ap, sizeof(Apuesta), 1, fapuestas))
    if ((j = acertados(ap, premiado, acc)) > 2)
    {
        fprintf(fpremiados, "%s %u: ", ap.id, j);
        for (i = 0; i<6; i++)
            fprintf(fpremiados, "%2u ", ap.n[i]);
        fprintf(fpremiados, "| ");
        for (i = 0; i<j; i++)
            fprintf(fpremiados, "%2u ", acc[i]);
        fprintf(fpremiados, "\n");
    }

fclose(fpremiados);
fclose(fapuestas);
return 0;
}
```