Современные методы и технологии программирования

Лекция 11

Случайные числа

Работа с файлами

Случайные числа

```
#include <time.h>
#include <stdlib.h>
int x = A + rand() % (B+1-A); //целое число из интервала [A..B]
```

Чтобы получать **разные** последовательности следует задавать начальный параметр (зерно)

```
srand( число );
```

Чтобы повторить **одну и ту же** последовательность случайных чисел два раза:

```
srand(NBEG); x=rand(); ... ; x=rand();
srand(NBEG); x=rand(); ... ; x=rand();
```

Реализация:

```
static unsigned long int next = 1L;
int rand(){
  next = next * 1103515245 + 12345;
  return ((unsigned int)(next/65536) % 32768);
}
void srand(unsigned int seed)
{     next = seed;     }

char t[sizeof(long)];
time(t); srand(t[0] + t[1] + t[2] + t[3] + getpid());
```

Работа с файлами

```
#include <stdio.h>
FILE *fopen ( const char *filename, const char* mode );
int fclose ( FILE *stream);
```

Форматный ввод и вывод

```
int fscanf (FILE* stream, const char* format,...);
int fprintf ( FILE* stream, const char* format,...);
```

Работа с символами

```
int fgetc (FILE *stream)
int fputc ( int c, FILE * stream)
int ungetc(int c, FILE *stream); //один символ
```

Работа со строками

```
char *fgets (char *s, int n, FILE *stream)
int fputs (const char *s, FILE * stream)
```

Позиционирование в файле

```
int fseek (FILE *stream, long offset, int origin)
long ftell (FILE *stream)
```

Бинарные файлы

```
#include <stdio.h>
FILE *fopen ( const char *filename, const char* mode );
int fclose ( FILE *stream);


int fread(void * p, unsigned sizelem, unsigned n, FILE *fp);
/* Возвращает количество полностью прочитанных элементов.
 В случае ошибки или EOF возвращаемое значение будет меньше n.
*/


size_t fwrite(const void * p, size_t size, size_t n, FILE * fp);
/* Возвращает количество полностью записанных элементов.
 В случае ошибки возвращаемое значение будет меньше n.
*/
```

Задачи

1. Программа. Определить, сколько раз в данном файле f встречается символ ‘A’. Имя файла берется из командной строки
2. По заданному файлу построить его копию, имена файлов задаются в командной строке. (Рассмотреть посимвольное копирование и копирование блоками)
3. Написать логическую функцию, которая проверяет, имеют ли два файла равное содержимое.
4. Файл содержит только символы-цифры, увеличить их значения на 1 за один просмотр файла (цифру '9' заменить на '0')

5. Программа. Определить, какая строка является самой длинной в заданном файле. Если таких строк несколько, то в качестве результата выдать первую из них. Имя файла задается в командной строке.

6. Программа из нескольких модулей: обработка матриц.
-- ввод значений матрицы из текстового файла,
-- распечатку матрицы в файл,
-- сохранение матрицы в **бинарный** файл,
-- загрузку матрицы из **бинарного** файла

Задачи (продолжение)

7. Программа — «архиватор» файлов.
Написать две функции: archive_s(s,t) и extract_s(s,t), параметр s - исходный файл, t - файл в которой требуется получить результат.

archive_s(s,t)

Последовательность из n одинаковых символов первого файла переносится во второй как 2 байта: байт, содержащий значение n и байт, содержащий код символа. Например, последовательность bbbbbb (занимает 6 байтов) должна быть заменена на 6b (2 байта), последовательность a заменяется на 1a. Повторяющиеся последовательности из более, чем 255 байтов поделить на несколько подпоследовательностей.

extract_s(s,t) — обратная операция

Сравнить размеры файлов, привести пример, когда «архивация» уменьшает размер файла, и когда происходит увеличение размера.