Занятие 5

Строки и фалы

Ввод строки 1

```
int main() {
	const int str_len=80;
	char s1[str_len]="This is a string";
	char s2[]="This is a string too";
	char *sconst= "This is a constant string";

	cout << "Enter a string: " << endl;
	cin >> s1;
	cout << s1 << endl;

	char s3[str_len], s4[str_len], s5[str_len];
	cout << "Enter a string: " << endl;
	cin >> s3 >> s4 >> s5;
	cout << s3 << " " << s4 << " " << s5 << endl;
	return 0;
}
```

Ввод строки 2

```
char t[str_len];
        cout << "Enter a string: " << endl;
        cin.getline(t, str_len);
        cout << t << endl;
```

Getline заменяет \n на \0
Метод get(s, n) оставляет \n в потоке; но вызов get() очищает поток

Чтение текстового файла

```
int i=1;
const int maxl=100;
char line[maxl];
ifstream fin("text.txt");
if (!fin) {
	cout << "File not found" << endl;
	return 1;
}

while (fin.getline(line, maxl)) {
	cout << "String nember " << i << ": " << line << endl;
	i++;
}
```

string.h

void *memcpy(void *dest, const void *src, size_t n);	копирует n байт из области памяти src в dest, которые не должны пересекаться, в противном случае результат неопределен (возможно как правильное копирование, так и нет)
void *memmove(void *dest, const void *src, size_t n);	копирует n байт из области памяти src в dest, которые в отличие от memcpy могут перекрываться
void *memchr(const void *s, char c, size_t n);	возвращает указатель на первое вхождение c в первых n байтах s, или NULL, если не найдено
int memcmp(const void *s1, const void *s2, size_t n);	сравнивает первые n символов в областях памяти
void *memset(void *, int z, size_t);	заполняет область памяти одним байтом z
char *strcat(char *dest, const char *src);	дописывает строку src в конец dest
char *strncat(char *dest, const char *src, size_t);	дописывает не более n начальных символов строки src (или всю src, если ее длина меньше) в конец dest
char *strchr(const char *, int);	ищет символ в строке, начиная с головы и возвращает его адрес, или NULL если не найден
char *strrchr(const char *, int);	ищет символ в строке, начиная с хвоста и возвращает его адрес, или NULL если не найден
int strcmp(const char *, const char *);	лексикографическое сравнение строк (возвращает "0" если строки разницу не имеют и "1", если разница есть)
int strncmp(const char *, const char *, size_t);	лексикографическое сравнение первых n байтов строк
int strcoll(const char *, const char *);	лексикографическое сравнение строк с учетом локали collating order
char *strcpy(char *toHere, const char *fromHere);	копирует строку из одного места в другое
char *strncpy(char *toHere, const char *fromHere, size_t n);	копирует до n байт строки из одного места в другое
char *strerror(int);	возвращает строковое представление сообщения об ошибке errno (не потоко-безопасная)
size_t strlen(const char *);	возвращает длину строки
size_t strspn(const char *s, const char *accept);	определяет максимальную длину начальной подстроки, состоящей исключительно из байтов, перечисленных в accept
size_t strcspn(const char *s, const char *reject);	определяет максимальную длину начальной подстроки, состоящей исключительно из байтов, не перечисленных в reject
char *strpbrk(const char *s, const char *accept);	находит первое вхождение любого символа, перечисленного в accept
char *strstr(const char *haystack, const char *needle);	находит первое вхождение строки needle в haystack
char *strtok(char *, const char *);	преобразует строку в последовательность токенов. Не потоко-безопасная, нереентрантная.
size_t strxfrm(char *dest, const char *src, size_t n);	создает оттранслированную копию строки, такую, что дословное сравнение ее (strcmp) будет эквивалентно сравнению с коллатором.

http://www.cplusplus.com/referenc
e/cstring/