str – пример переменной

не забыть #include <string> !!!!!

1. ***strcpy***(*S*1, *S*2) – копирует содержимое строки *S*2 в строку *S*1;
2. ***strncpy –*** *копирование n символов;*
3. ***strcat***(*S*1, *S*2) – присоединяет строку *S*2 к строке *S*1 и помещает ее в массив, где находилась строка *S*1, при этом строка *S*2 не изменяется; нулевой байт, который завершал строку *S*1, заменяется первым символом строки *S*2; можно просто в новой строке написать (str3 = str1 + str2);
4. ***strncat –*** *перенос n символов;*

3) ***strcmp***(*S*1, *S*2) – сравнивает строки *S*1 и *S*2 и возвращает значение =0, если строки равны, т.е. содержат одно и то же число одинаковых символов; значение <0, если *S*1<*S*2; и значение >0, если *S*1>*S*2. ;

4) ?***strlen(char)?***(*S*) – возвращает длину строки, т.е. количество символов, начиная с первого (*S*[0]) и до нуль-терминатора, который не учитывается. Чтобы узнать длину строки, можно воспользоваться функцией-членом size(), или length(), которые, как и ожидается, возвращают саму длину (длина не включает завершающий нулевой символ).`{str.length/.size}(only for string).;

5) ***strstr*(***S*1, *S*2**) –** указывает первое появление подстроки *S*2 в строке *S*1;

6) преобразование строки *S* в число (библиотека *stdlib*.*h*): целое – (int) ***atoi***(*S*); длинное целое – (long) ***atol***(*S*); действительное – (double) ***atof***(*S*);

\*при возникновении ошибки данные функции возвращают значение 0\*

7) преобразование числа *V* в строку *S*: целое – ***itoa***(int *V*, char *S*, int *kod*); длинное целое – ***ltoa***(long *V*, char *S*, int *kod*); значение параметра *kod* определяет выбор системы счисления для преобразования и находится в диапазоне 2≤*kod*≤36, для отрицательных чисел *kod* = 10;

8) empty(), возвращающий true для пустой строки и false для непустой: cheking for fulling of string;

9) cin.getline/getline – ввод строки с пробелами;