**NOTES**

* Buffer is not null terminated whereas string is.

Char a[]={‘y’,’a’,’s’,’h’} //buffer

Char b[]={‘y’,’a’,’s’,’h’,’/0’}//string

* Char str[]=”hello”-> {h,e,l,l,o,\0}; requires 6 memory space
* Char ch=”h”->{h,\0}; requires 2
* Char ch=’h’->{h} requires 1
* Char ch=’ ’ ->{\0} c++ allows empty string but not empty character
* Char ch[5]=”hello” ,null character is not appended coz hello occupies all the space..for null character it should be ch[6],5 memory space for hello and 1 for ‘\0’
* Sizeof gives number of characters+1,length gives number of character
* In reality,ascii codes of character are stored in memory in place of character itself
* Char str[]=”hello”; char str[]={‘h’, ‘e’, ‘l’, ‘l’, ‘o’, ‘\0’}; char[10]=”hello”-> {h,e,l,l,o,\0,\0,\0,\0,\0};
* Char str[3];

Str=”hello”; //illegal stament

* Char str[]; //illegal statement..array size should be given
* Reading strings:

1)cin-enters a single sting,space is a delimiter i.e terminates as soon as it finds a blank space

>>getche,getchar ( functions ..stdio.h) enters single character and are used to hold output screen.

>>gets(c function,enter complete line)

2) getch->characters are not displayed on the screen when entering. Header file->conio.h

3)getche-character is displayed without pressing enter

4)getchar-charater is displayed after pressing enter

5)getline(c++ function)->used to enter a complete line..delimiter is new line..used as

>> cin.getline(b,10); only first 10 characters will be stored in string b..irrespective of any number of characters entered.

>> cin.getline(b,’\*’); only that characters will be stored in string b which are entered before \*.

* The difference between getline() and gets() is that getline is c++ function and gets is c function.gets() has newline as delimiter ,same as getline.
* gets,cin,getline automatically adds null character at the end of line;
* writing strings:

>>c++ functions:

1. cout (ouputs complete line but doesnot terminate the line with new line i.e. continues with same line for new output)
2. cout.put(b[1]) (gives single character on output)

>>c fuctions:

Requires stdio.h headerfile

1. puts (ouputs complete line and also terminates the line with new line i.e. goes to new line for new output).

Also return EOF if an error occurs and positive number on success.

1. Putch and putchar: returns single character

//implement this for concept clarity

char a[5];

char b[100];

int i=0;

char ch=cout.get();

while(ch!='\*')

{ a[i]=ch;

i++;

ch=cout.get();

}a[i]='\0'; cin.getline(b,10);

cin.getline(a,5);

gets(b);

/puts(b);

puts(a);

* There is a difference between usage of a character array and the string(sequence of characters);

Char str[10];

gets(str); {**‘h’,’e’,’l’,’l’,’o’,’\0’,**-,-,-,-}

>>underlined part is string while rest is part of array ,not string.

* Character manipulation functions:

isalpha(),isdigit(),isalnum(),tolower(),toupper()

islower(),isupper(),isspace(),

ispunct(punctuation marks,?-!(),’,!;)

* All of them return int value,some return ascii value

Example:-tolower(‘A’) it will return ascii value of a=97.Alphanumeric includes:-digit from 0-9 and

a-z,A-Z;

* Ascii value: {a=97,z=123},{A=65,z=91},{0=48,9=57}

**STRINGS FUNCTIONS DEFINED UNDER STRING.H**

* **strcat(char \*dest,const char \*src)**

char src[] or char \*src=”singla” //both valid

char dest[]=”yash”; //if we store yash in \*dest..there

strcat(dest,src); // will be segmentation fault prog-crash

//a terminating null character is appended to dest

//before returning to calling function.

* **strchr(const char \*a,int c):** searches first occurrence of unsigned char c in string “a” from starting of string and returns pointer pointing to first matching character

char a[] or char \*a=”yash”; //both valid

char \*pos = strchr(a,’y’);

//return null if character not found

* **strrchr(const char \*a,int c):** same as above but searches first occurrence from back.
* **strncmp(const char \*a,const char \*b,int n):**

compares first n characters of a and b strings;

* **strcpy( char \*dest,const char \*src):**

char src[] or char \*src=”singla” //both validchar dest[]=”yash”; //if we store yash in \*dest..there strcpy(dest,src); // will be segmentation fault.. prog-crash

//copies src to dest along with null character

//imp thing is…size of dest should be large enough to store src string acc to rules but in real practice if dest size is shorter ..src will overwrite some memory:buffer overflow..and if we print dest,result is perfect.

* **strncpy( char \*dest,const char \*src,n):**

copies first n char from src to dest..n char may includes null character…if n=0 nothing is copied

and if n is negative..it may be a segmentation fault and program gets crashed.

* Use **strncpy instead of stcpy** strncpy combats buffer overflow by requiring you to put a length in it.Also length in strncpy can be used in a way like:

>>  **strncpy(dest,src,sizeof(dest)-1);** this will enforce copying of only that much characters of src for which dest has space. -1 is for null character

* Strlen(str):count length of string..excluding null character.
* Strstr( char\* a, char \*b):searches first occurrence of b in a (doesnot search null character).returns pointer to first occurrence of b in a;

>>all of following functions are included in stdlib.h

* atoi(const char \*str):converts given string passed as an argument to integer. string should start with an integer and will stop reading string as soon as it finds non-numeric character. Example:

i=atoi(1234.567);

result:i=1234

* atof(const char \*str)->string to double

example: atof(“12.45 is yash”),out:12.45

* atol(const char \*str)->string to long
* example: atol(“12.45 is yash”),out:12.45L
* itoa(int value,char \*str,base):integer to string

base:10 for decimal,16 for hex,8octal,2 binary

//itoa() is of great use in converting decimal to binary for using in programs