Day 14

STREAMS

input- data moved from external location to RAM

stream-sequence of characters. text or binary streams. text streams organized into lines up to 255 characters long and terminated with newline

5 standard streams:

```
1. stdin- standard input
2. stdout- standard output
3. stderr- stadard error
4. stdprn- standard printer
5. stdaux- standard auxillary
```

each stream connected to file- intermediate step between stream that program deals with and actual device being used for input or output

Accepting Keybaord Input

character input, line input, formatted input

Character Input

read input from stream one character at a time.

some buffered- holds in storage space until you press enter then sends to stdin stream

some automatically echo each character as received

getchar()- obtains next character from stream stdin. provides buffered character input with echo

no characters are received until you press enter. each keypress assigned to variable if you choose. only gets 1

getch()- unbuffered input without echo. returns each character as soon as key pressed. does not print to screen. only gets 1

buffered character functions translate \r- carriage return to \n-newline. unbuffered do not

getche()- like getch() but echoes

getc() and fgetc()- don't automatically work with stdin let program specify input stream

fgets()- reads line of text from an input stream, can specify how output.

char *fgets(char *str, int n, FILE *fp);

```
1. char \*str where stored
2. int n - max characters to input. nothing specified will
read until new line, or eof
3. specify input stream
```

Formatted input

scanf() and fscanf()

use [^*character*] to truncate strings **scanf()-** buffered. pg 351 for gee-wiz modifiers. extra characters can wait in stdin need to use fflush(stdin) to clear

Controlling Output

character output, line output, formatted output

putchar()- sends single character to stdout. will accept int but will print ascii value of int

fputc()- int fputc(int c, FILE *fp)

puts()- int puts(char *cp). displays string up to null

printf()- see pg 363 for gee-wiz modifiers.

Redirection

UNIX redirection < > >> work:

redirects standard input. redirect.c > input.txt changes standard input to input.txt

redirect.c > test.txt

stderr always connected to screen