1. Describe strings
   1. Any number with a + or – sign infront and adding decimal
      1. +34324.35234
      2. -2321.5534
      3. 1231245
   2. Atleast one letter followed by a space, then 5 then zero or one 5, space then and any number of letters
      1. Fppoasdjf 5 oijfe
      2. Ijdij 5
      3. Billy 55 mac
2. Build a Regex
   1. \\_(\w+\d)\*
   2. \(?[0-9]{3}\)?\-?[0-9]{3}-[0-9]{4}
   3. v00[0-9]{6}
3. There are 720 combinations of different combinations and it cannot work with no spaces. It cannot work because the separation of the articles nouns and verbs will not be the same without spaces.

Translation-unit -> external-declaration

External-declaration -> function-definition

Function-definition -> [declaration-specifiers] declarator

[declaration-list] compound-statement

Declaration -> [declaration-specifiers] ‘;’

Init-declarator-list -> init-declarator

init-declarator -> declarator

declarator -> [pointer] direct-declarator

pointer -> [‘\*’] type-qualifier-list pointer

direct-declarator -> ID

Expr -> term { + term} | exp

Exp -> Term

Term -> factor { \*factor} | mod

Mod -> factor | division

Factor -> (expr) | number |division

Division -> subtract | number

Subtract -> number

Number -> digit { digit}

Digit -> 0|1|2|3|4|5|6|7|8|9

1. Separate files, multiple pictures
2. Separate file
3. Sfd

line 6 X

line 7 116

line 12 b

line 13 5.6

line 17 t

#9

# include <stdio.h>

#include <stdlib.h>

#include <string.h>

gcd(int \*a, int \*b){

int r = \*b;

while(r != 0){

\*b = r;

r = \*a % \*b;

\*a = \*b;

}

}

main() {

char str[999];

FILE \* file;

file = fopen( "chap6\_7File.txt" , "r");

if (file) {

while (fgets(str, 255,(FILE\*)file)!= NULL){

char \*value = strtok(str, " ");

int aval = atoi(value);

value = strtok(0, " ");

int bval = atoi(value);

gcd(aval, bval);

printf(\*b);

}

fclose(file);

}

}