/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 1학년 1학기 프로젝트 : 큰수계산기 \*\*\*

\*\*\* 시작 날짜 : 2016년5월 29일 \*\*\*

\*\*\* 팀원 : 박순욱(20160290) \*\*\*

\*\*\* 임규형(20160318) \*\*\*

\*\*\* 전혜윤(20160326) \*\*\*

\*\*\* 배서현(20160294) \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define N 10

void starter(char input[]); //수식 입력 함수

void calculate(char value[], char answer[]); //수식 계산 함수

int compare(char a[], int v1, char b[], int v2); //두 배열 크기 비교

int plus(char a[], int n, char b[], int m, char answer[]); //덧셈 함수

int minus(char a[], int n, char b[], int m, char answer[]); //뺄셈 함수

int multiply(char a[], int n, char b[], int m, char answer[]); //곱셈 함수

int divide(char a[], int n, char b[], int m, char answer[]); //나눗셈 함수

int mod(char a[], int n, char b[], int m, char answer[]); //나머지 함수

void copy(char a[],char b[]); //복붙

void multiplyTen(char a[]); //10배 함수

void variable(char varName[], char value[], int length, int negCheck, char deciCount[], int saveVariable[], int checkVar[]);//변수 정의 함수

void VAR(int checkVar[], int saveVariable[]); //VAR 함수

void save(int checkVar[], int saveVariable[]); //save 함수

void load(int checkVar[], int saveVariable[]); //load 함수

void integer(char inte[], int n); //받은 배열(상수)을 char형에서 int형으로 바꾸는 함수

void reverse(char rev[], int n); //받은 배열(상수)을 뒤집어 주는 함수

void printanswer(char answer[], int length, int sigDigit, int negNum); //정답 출력 함수

//변수 관련 배열

static int saveVariable[26][62] = {0};

static int checkVar[26] = {0};

//메인 함수

int main()

{

while(1){

//처음 수식을 받을 배열

char input[500] = {0};

//수식 입력 함수 호출

starter(input);

//end 처리

if(input[0]=='e' && input[1]=='n' && input[2]=='d' && input[3]=='\0'){

break;

}

//clear이면 함수 종료

else if(input[0]=='c' && input[1]=='l' && input[2]=='e' && input[3]=='a' && input[4]=='r' && input[5]=='\0'){

system("clear");

}

}

return 0;

}

//수식 계산 함수

void starter(char input[])

{

int c, i=0;

//답 저장 배열

char answer[60] = {0};

printf("입력 : ");

//처음 입력 받기

while(1){

c = getchar();

//수식 입력의 마지막 처리

if(c=='\n'){

input[i] = '\0';

break;

}

//띄어쓰기 무시

else if((c==' ') || (c==' ')){

continue;

}

//받을 수 있는 것(숫자, 알파벳, 수식 기호, 소수점)은 input배열에 저장

else if(((c>='0') && (c<='9')) || ((c>='a') && (c<='z')) || ((c>='A') && (c<='Z')) || (c=='+') || (c=='-') || (c=='\*') || (c=='/') || (c=='%') || (c=='=') || (c=='.')){

input[i] = c;

i++;

}

//기타 문자 오류 처리

else{

printf("error : '%c'는 인식할 수 없습니다\n\n",c);

input[i] = c;

while(1){

c = getchar();

if(c=='\n'){

break;

}

}

return;

}

}

//end이면 함수 종료

if(input[0]=='e' && input[1]=='n' && input[2]=='d' && input[3]=='\0'){

return;

}

//clear이면 함수 종료

else if(input[0]=='c' && input[1]=='l' && input[2]=='e' && input[3]=='a' && input[4]=='r' && input[5]=='\0'){

return;

}

//save이면 함수 종료

else if(input[0]=='s' && input[1]=='a' && input[2]=='v' && input[3]=='e' && input[4]=='\0'){

save(checkVar, saveVariable);

printf("notice : 정의된 변수들을 저장합니다(파일 이름 : VAR)\n\n");

return;

}

//load이면 함수 종료

else if(input[0]=='l' && input[1]=='o' && input[2]=='a' && input[3]=='d' && input[4]=='\0'){

load(checkVar, saveVariable);

printf("notice : VAR 파일에서 변수들을 불러옵니다\n\n");

return;

}

//VAR이면 함수 종료

else if(input[0]=='V' && input[1]=='A' && input[2]=='R' && input[3]=='\0'){

VAR(checkVar, saveVariable);

return;

}

//수식 계산 함수 호출

calculate(input, answer);

return;

}

//수식 계산 함수

void calculate(char input[], char answer[])

{

int c, i=0, j, k, l, temp;

int v1, v2;

int length, sigDigit=0, negNum=0, negCheck=0;

//수와 문자와 기호를 저장할 배열

char value[N][60] = {0};

//소수

char deciCount[N][2] = {0};

//수식 구분하기

if(input[0]=='-'){

negCheck+=1;

i++;

}

for(j=0, k=0; i<strlen(input); i++){

c = input[i];

//숫자

if((c>='0') && (c<='9')){

if(k>=1){

if(((value[j][k-1]>='a') && (value[j][k-1]<='z')) || ((value[j][k-1]>='A') && (value[j][k-1]<='Z'))){

printf("error : 수식을 잘못 입력하셨습니다\n\n");

return;

}

}

//소수 카운트

if(deciCount[j][0]){

deciCount[j][1]+=1;

}

value[j][k] = c;

k++;

}

//알파벳

else if(((c>='a') && (c<='z')) || ((c>='A') && (c<='Z'))){

if(k>=1){

if((((value[j][k-1]>='0') && (value[j][k-1]<='9')) || deciCount[j][0]) || (((value[j][k-1]>='a') && (value[j][k-1]<='z')) || ((value[j][k-1]>='A') || (value[j][k-1]<='Z')))){

printf("error : 수식을 잘못 입력하셨습니다\n\n");

return;

}

}

value[j][k] = c;

k++;

}

//수식기호

else if((c=='+') || (c=='-') || (c=='\*') || (c=='/') || (c=='%') || (c=='=')){

if(c=='-' && j==2){

negCheck += 2;

}

//전에 배열이 상수라면 소수자리 채우기

else{

if(value[j][0]>='0' && value[j][0]<='9'){

for(l=deciCount[j][1]; l<9; l++){

value[j][k]='0';

k++;

}

}

value[j][k]='\0';

value[j+1][0] = c;

j+=2;

k=0;

}

}

//소수점

else if (c=='.'){

if(deciCount[j][0]){

printf("error : 소수를 잘못 입력하셨습니다\n\n");

return;

}

deciCount[j][0]=1;

}

//배열의 끝

else if(c=='\0'){

value[j][k]='\0';

break;

}

//소수 최대 자리를 초과했는지 검사

if(deciCount[j][0]){

if(deciCount[j][1]==10){

printf("error : 소수부분은 9자리까지만 입력해주세요\n\n");

return;

}

}

//정수 최대 자리를 초과했는지 검사

else{

if(k==51){

printf("error : 정수부분은 50자리까지만 입력해주세요\n\n");

return;

}

}

}

//마지막 배열의 소수자리 처리

if(value[j][0]>='0' && value[j][0]<='9'){

for(l=deciCount[j][1]; l<9; l++){

value[j][k]='0';

k++;

}

}

v1=strlen(value[0]);

v2=strlen(value[2]);

if(value[1][0]=='+' || value[1][0]=='-' || value[1][0]=='\*' || value[1][0]=='/' || value[1][0]=='%'){

for(i=0;i<3;i+=2){

if(((value[i][0]>='A') && (value[i][0]<='Z')) || ((value[i][0]>='a') && (value[i][0]<='z'))){

if(value[i][0]>='a'){

temp=value[i][0]-'a';

}

else{

temp=value[i][0]-'A';

}

//변수가 저장되있는지 확인

if(checkVar[temp]){

//전체 자리수 저장

length = saveVariable[temp][59]-saveVariable[temp][61]+9;

//변수 출력

for(j=0;j<length;j++){

if(saveVariable[temp][j]!=0){

value[i][j] = saveVariable[temp][j]+'0';

}

else{

value[i][j] = '0';

}

}

//부호 저장

negCheck += saveVariable[temp][60];

//소수자리 저장

if(saveVariable[temp][61]){

deciCount[i][0] = 1;

deciCount[i][1] = saveVariable[temp][61];

}

if(i){

v2=length;

}

else{

v1=length;

}

}

else{

printf("error : %c는 선언되지 않았습니다\n\n",value[i][0]);

return;

}

}

}

}

//'+'일 경우

if(value[1][0]=='+'){

if(deciCount[0][1]>=deciCount[2][1]){

sigDigit = deciCount[0][1];

}

else{

sigDigit = deciCount[2][1];

}

if(negCheck==0){

length = plus(value[0],v1,value[2],v2,answer);

}

else if(negCheck==1){

if(compare(value[0],v1,value[2],v2)<2){

length = minus(value[0],v1,value[2],v2,answer);

}

else{

length = minus(value[2],v2,value[0],v1,answer);

}

if(compare(value[0],v1,value[2],v2)==1){

negNum = 1;

}

}

else if(negCheck==2){

if(compare(value[0],v1,value[2],v2)<2){

length = minus(value[0],v1,value[2],v2,answer);

}

else{

length = minus(value[2],v2,value[0],v1,answer);

}

if(compare(value[0],v1,value[2],v2)==2){

negNum = 1;

}

}

else if(negCheck==3){

length = plus(value[0],v1,value[2],v2,answer);

negNum = 1;

}

reverse(answer,length);

}

//'-'일 경우

else if(value[1][0]=='-'){

if(negCheck==0){

if(compare(value[0],v1,value[2],v2)<2){

length = minus(value[0],v1,value[2],v2,answer);

}

else{

length = minus(value[2],v2,value[0],v1,answer);

}

if(compare(value[0],v1,value[2],v2)==2){

negNum = 1;

}

}

else if(negCheck==1){

length = plus(value[0],v1,value[2],v2,answer);

negNum = 1;

}

else if(negCheck==2){

length = plus(value[0],v1,value[2],v2,answer);

}

else if(negCheck==3){

if(compare(value[0],v1,value[2],v2)<2){

length = minus(value[0],v1,value[2],v2,answer);

}

else{

length = minus(value[2],v2,value[0],v1,answer);

}

if(compare(value[0],v1,value[2],v2)==1){

negNum = 1;

}

}

reverse(answer,length);

if(deciCount[0][1]>=deciCount[2][1]){

sigDigit = deciCount[0][1];

}

else{

sigDigit = deciCount[2][1];

}

}

//'\*'일 경우

else if(value[1][0]=='\*'){

length = multiply(value[0],v1,value[2],v2,answer);

reverse(answer,length);

if(negCheck==1 || negCheck==2){

negNum = 1;

}

if(deciCount[0][1]+deciCount[2][1]>=9){

sigDigit = 9;

}

else{

sigDigit = deciCount[0][1]+deciCount[2][1];

}

}

//'/'일 경우

else if(value[1][0]=='/'){

length = divide(value[0],v1,value[2],v2,answer);

int a = length;

while(answer[--a] == 0);

sigDigit = 10-(length -a);

if(negCheck==1 || negCheck==2){

negNum = 1;

}

}

//'%'일 경우

else if(value[1][0]=='%'){

length = mod(value[0],v1,value[2],v2,answer);

int a = length;

while(answer[--a] == 0);

sigDigit = 10-(length -a);

if(negCheck==1 || negCheck==3){

negNum=1;

}

}

//'='일 경우

else if(value[1][0]=='='){

if(((value[0][0]>='A') && (value[0][0]<='Z')) || ((value[0][0]>='a') && (value[0][0]<='z'))){

v2=strlen(value[2]);

variable(value[0], value[2], v2, negCheck, deciCount[2], saveVariable, checkVar);

length = v2;

if(negCheck>1){

negNum = 1;

}

sigDigit = deciCount[2][1];

for(i=0;i<v2;i++){

answer[i] = value[2][i];

}

}

}

//변수를 불러오는 경우

else if(((value[0][0]>='A') && (value[0][0]<='Z')) || ((value[0][0]>='a') && (value[0][0]<='z'))){

if(value[0][0]>='a'){

temp=value[0][0]-'a';

}

else{

temp=value[0][0]-'A';

}

//변수가 저장되있는지 확인

if(checkVar[temp]){

//전체 자리수 저장

length = saveVariable[temp][59]-saveVariable[temp][61]+9;

//변수 출력

for(i=0;i<length;i++){

answer[i] = saveVariable[temp][i];

}

//부호 저장

negNum = saveVariable[temp][60];

//소수자리 저장

sigDigit = saveVariable[temp][61];

}

else{

printf("error : %c는 선언되지 않았습니다\n\n",value[0][0]);

return;

}

}

else{

printf("error : 잘못된 수식입니다\n\n");

return;

}

//정답 출력 함수 호출

printanswer(answer,length,sigDigit,negNum);

printf("\n\n");

return;

}

//두 배열 비교 함수

int compare(char a[], int v1, char b[], int v2){

int i;

//앞이 크면 1을 리턴한다

if(v1 > v2)

return 1;

//뒤가 크면 2를 리턴한다

else if(v1 < v2)

return 2;

else

for(i=0 ; i < v1; i++){

if(a[i] > b[i])

return 1;

else if(a[i] < b[i])

return 2;

}

//같으면 0을 리턴한다

return 0;

}

//덧셈 함수

int plus(char a[], int n, char b[], int m, char answer[])

{

int i, j;

//받은 배열(상수)를 char형에서 int형으로 바꾼다

integer(a,n);

integer(b,m);

//받은 배열(상수)을 뒤집는다

reverse(a,n);

reverse(b,m);

//전체 자리를 위한 수 리턴

if(n>=m){

j=n;

}

else{

j=m;

}

//두 수를 한자리씩 더해준다

for(i=0; i<j; i++){

answer[i] = a[i] + b[i];

}

//10이 넘어가면 올려준다

for(i=0; i<j; i++){

if(answer[i] >= 10){

answer[i] -= 10;

answer[i+1] += 1;

}

}

if(answer[i]){

j += 1;

}

return j;

}

//뺄셈 함수

int minus(char a[], int n, char b[], int m, char answer[])

{

int i,j;

j=n;

//받은 배열(상수)를 char형에서 int형으로 바꾼다

integer(a,n);

integer(b,m);

//받은 배열(상수)을 뒤집는다

reverse(a,n);

reverse(b,m);

//두 수를 한자리씩 빼준다

for(i=0;i<n;i++){

answer[i] = a[i] - b[i];

}

//0보다 작으면 10을 더해준다

for(i=0; i<=n; i++){

if(answer[i] < 0){

answer[i] += 10;

answer[i+1] -= 1;

}

}

//전체 자리를 위한 수 리턴

for(i=n; i>n-m+9; i--){

if(answer[i-1]!=0){

break;

}

else if((answer[i-1]==0) && (i!=10)){

j -= 1;

}

}

return j;

}

//곱셈 함수

int multiply(char a[], int n, char b[], int m, char answer[])

{

int i,j;

int carry=0;

char temp[60][120];

for(i=0;i<60;i++){

for(j=0;j<120;j++){

temp[i][j] = 0;

}

}

//받은 배열(상수)를 char형에서 int형으로 바꾼다

integer(a,n);

integer(b,m);

//받은 배열(상수)을 뒤집는다

reverse(a,n);

reverse(b,m);

//곱셈을 한자릿수마다 연산한다

for(j=0;j<m;j++){

for(i=0;i<n;i++)

{

temp[j][i] = (a[i] \* b[j] + carry)%10;

carry = (a[i] \* b[j] + carry)/10;

}

temp[j][i] = carry;

carry = 0;

}

//자릿수를 맞추기 위해 밀어준다

for(j=1;j<m;j++)

{

for(i=n;i>=0;i--)

temp[j][i+j] = temp[j][i];

}

//밀어준 자리에 0을 넣는다

for(j=1;j<=m;j++)

{

for(i=1;i<=j;i++)

temp[j][j-i] = 0;

}

for(j=0;j<m;j++){

for(i=0;i<m+n;i++)

{

if(carry +temp[0][i] + temp[j+1][i] >= 10)

{

temp[0][i] = (temp[0][i] + temp[j+1][i] + carry)%10;

carry=1;

}

else

{

temp[0][i] = temp[0][i] + temp[j+1][i] + carry;

carry=0;

}

}

}

//전체 자리를 위한 수를 리턴한다

j=m+n;

for(i=0;i<j;i++){

temp[0][i]=temp[0][i+9];

}

j-=9;

for(i=m+n-9;i>10;i--){

if(temp[0][i-1]==0){

j-=1;

}

else{

break;

}

}

if(j>=60){

j=59;

}

for(i=0;i<j;i++){

answer[i] = temp[0][i];

}

return j;

}

//나눗셈 함수

int divide(char a[], int n, char b[], int m, char answer[])

{

char ta[60];

char tb[60];

char tbb[60];

int j;

int FLOAT = 9; //상수

copy(ta,a);

copy(tb,b);

int howBig = 0;

char temp[60];

copy(temp,b);

multiplyTen(temp);

while(compare(ta,strlen(ta),temp,strlen(temp)) <= 1){

howBig++;

copy(tb,temp);

multiplyTen(temp); //tb \*= 10;

}

copy(tbb,tb);

j = howBig;

while(howBig != -10){

int count = 0;

char getAnswer[60] = {0};

while(compare(ta,strlen(ta),tb,strlen(tb)) == 1){

int len = minus(ta,strlen(ta),tb,strlen(tb),getAnswer);

for(int i = 0; i < len / 2;i++){

int t = getAnswer[i];

getAnswer[i] = getAnswer[len - i - 1];

getAnswer[len - i - 1] = t;

}

for(int i = 0; i < len; i++)

getAnswer[i] += '0';

copy(ta,getAnswer);

copy(tb,tbb);

count++;

}

if(compare(ta,strlen(ta),tb,strlen(tb)) == 0){

count ++;

answer[FLOAT + howBig] = count;

break;

}

answer[FLOAT + howBig] = count;

howBig--;

multiplyTen(ta);

}

j += FLOAT + 1;

for(int i = 0; i < j / 2; i++){

int t = answer[i];

answer[i] = answer[j - i - 1];

answer[j - i - 1] = t;

}

//전체 자리수 리턴

return j;

}

//나머지 함수

int mod(char a[], int n, char b[], int m, char answer[]){

char ta[60];

char tb[60];

char temp[60];

copy(ta,a);

copy(tb,b);

char getAnswer[60] = {0};

int len,i,t;

len = divide(ta,strlen(ta),tb,strlen(tb),getAnswer);

for(i = 0; i < 9; i++)

getAnswer[len - i - 1] = 0;

for(i =0; i < len; i++)

getAnswer[i] += '0';

copy(temp,getAnswer);

len = multiply(temp,strlen(temp),tb,strlen(tb),getAnswer);

for(i = 0; i < len/2; i++){

t = getAnswer[i];

getAnswer[i] = getAnswer[len - i -1];

getAnswer[len - i - 1] = t;

}

for(i = 0; i < len; i++)

getAnswer[i] += '0';

copy(tb,getAnswer);

len = minus(ta,strlen(ta),tb,strlen(tb),answer);

for(i = 0; i < len / 2; i++){

t = answer[i];

answer[i] = answer[len - i - 1];

answer[len - i - 1] = t;

}

//전체 자리수 리턴

return len;

}

//배열 복사 함수

void copy(char a[], char b[]){

for(int i = 0; i < 60; i++)

a[i] = b[i];

}

//10배 함수

void multiplyTen(char a[]){

a[strlen(a)] = '0';

}

//변수 정의 함수

void variable(char varName[], char value[], int length, int negCheck, char deciCount[], int saveVariable[], int checkVar[]){

static int varCount = 0;

int i,j,temp;

integer(value, length);

if(varName[0]>='a'){

temp=varName[0]-'a';

}

else{

temp=varName[0]-'A';

}

//변수 개수 카운트

if(checkVar[temp]==0){

varCount++;

checkVar[temp] = 1;

}

//변수를 10개 이상 선언

if(varCount>10){

printf("error : 변수는 10개까지 입력 받을 수 있습니다.\n\n");

return;

}

//변수 저장

for(i=0;i<length;i++){

saveVariable[temp\*62+i] = value[i];

}

//소수자리 저장

saveVariable[(temp+1)\*62-1] = deciCount[1];

//부호 저장

if((negCheck==1) || (negCheck==2)){

saveVariable[(temp+1)\*62-2] = 1;

}

//전체 자리수 저장

saveVariable[(temp+1)\*62-3] = length-9+deciCount[1];

return;

}

//VAR 함수

void VAR(int checkVar[], int saveVariable[]){

int i,j;

for(i=0;i<26;i++){

//변수가 정의되어있는지 확인

if(checkVar[i]){

printf("%c = ", i+'A');

//변수 저장

if(saveVariable[(i+1)\*62-2]){

printf("-");

}

for(j=0;j<saveVariable[(i+1)\*62-3];j++)

{

//소수점 찍기

if(j==saveVariable[(i+1)\*62-3]-saveVariable[(i+1)\*62-1]){

printf(".");

}

printf("%d",saveVariable[i\*62+j]);

}

printf("\n");

}

}

printf("\n");

return;

}

//save 함수

void save(int checkVar[], int saveVariable[]){

FILE \*save;

int i,j;

save = fopen("VAR","w");

for(i=0;i<26;i++){

//변수가 선언되어있는지 확인

if(checkVar[i]){

fprintf(save,"%c = ", i+'A');

//변수 저장

if(saveVariable[(i+1)\*62-2]){

fprintf(save,"-");

}

for(j=0;j<saveVariable[(i+1)\*62-3];j++)

{

//소수점 찍기

if(j==saveVariable[(i+1)\*62-3]-saveVariable[(i+1)\*62-1]){

fprintf(save,".");

}

fprintf(save,"%d",saveVariable[i\*62+j]);

}

fprintf(save,"\n");

}

}

fclose(save);

return ;

}

//load 함수

void load(int checkVar[], int saveVariable[]){

FILE \*load;

int i,j,k,deci=0;

int name;

char temp[61] = {0};

load = fopen("VAR","r");

while((fscanf(load, "%c = %s\n",&name, &temp)!=EOF))

{

checkVar[name-'A'] = 1;

for(i=0,j=0,k=0,deci=0;i<strlen(temp);i++){

if(temp[i]>='0' && temp[i]<='9'){

saveVariable[(name-'A')\*62+j]=temp[i]-'0';

j++;

if(deci){

k++;

}

}

else if(temp[i]=='.'){

deci=1;

}

else if(temp[i]=='-'){

saveVariable[(name-'A'+1)\*62-2]=1;

}

}

saveVariable[(name-'A'+1)\*62-3]=j;

saveVariable[(name-'A'+1)\*62-1]=k;

}

fclose(load);

return;

}

//받은 배열(상수)을 char형에서 int형으로 바꾸는 함수

void integer(char inte[], int n)

{

int i;

for(i=0; i<n; i++){

if(inte[i]!=0){

inte[i] -= '0';

}

}:

return;

}

//받은 배열(상수)을 뒤집어 주는 함수

void reverse(char rev[], int n)

{

int tmp;

int i;

for(i=0; i<n/2; i++){

tmp = rev[i];

rev[i] = rev[n-1-i];

rev[n-1-i] = tmp;

}

return;

}

//정답 출력 함수

void printanswer(char answer[], int length, int sigDigit, int negNum)

{

int i, j;

printf("답 : ");

if(negNum){

printf("-");

}

for(i=0, j=0; i<length; i++){

//소수자리 출력

if(i==length-9){

if(sigDigit!=0){

printf(".");

for(j=0; i<length+sigDigit-9; i++){

printf("%d", answer[i]);

j++;

if((j%3==0) && j!=sigDigit){

printf(",");

}

}

return;

}

else{

return;

}

}

//콤마 출력

else if((j%3==length%3) && (j!=0 && length!=1)){

printf(",");

}

//정답 출력

printf("%d", answer[i]);

j++;

}

return;

}