학번: 2019202052

이름: 김호성

Test 1

#include<stdio.h> //표준입출력헤더파일 포함

int main() //main함수 생성

{

int i = 2, j = 4; //좌표 포인트 정하는 변수 생성

int x1, x2, y1, y2; //좌표 입력 받을 변수

int height, width, area; //가로 세로 넓이변수

int temp; //값 이동 변수

back: //입력받은 값에 오류가 있을경우 돌아옴.

printf("첫 번째 좌표를 입력해 주세요 : ");

scanf("%d %d", &x1, &y1); //첫 번째 좌표를 입력받음

printf("두 번째 좌표를 입력해 주세요 : ");

scanf("%d %d", &x2, &y2); //두 번째 좌표를 입력받음

if((x1 == x2) || (y1 == y2)) //x나 y값 중 같은 값이 있으면 직선이 되어버림.

{

printf("잘못된 좌표를 입력하셨습니다. \n\n");

goto back; //좌표 값 다시 받음

}

if((x1 < x2)&&(y1 < y2)) //좌표 포인트 정하는 조건

{

i = 1;

j = 3;

}

if(x1 > x2) //후에 가로에 음수 값이 안나오게 함.

{

temp = x1;

x1 = x2;

x2 = temp;

}

if(y1 > y2) //후에 세로에 음수 값이 안나오게 함.

{

temp = y1;

y1 = y2;

y2 = temp;

}

width = x2 - x1; //가로 계산

height = y2 - y1; //세로 계산.

area = width \* height; //넓이 계산

printf("첫번째 점의 좌표 point %d (%d,%d) \n",i, x1, y1); //첫 번째 좌표위치 출력

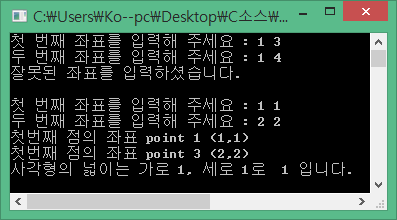
printf("첫번째 점의 좌표 point %d (%d,%d) \n",j, x2, y2); //두 번째 좌표위치 출력

printf("사각형의 넓이는 가로 %d, 세로 %d로 %d 입니다.\n", width, height, area); //넓이 출력

return 0; //main함수 종료

}

실행결과



Test2

#include <stdio.h> //표준입출력헤더파일 포함

int main() //main함수 생성

{

int num, n1, n2, n3, n4, n5; //입력받을 num과 n1~n5 생성

printf("0~50000중 숫자를 입력하시오. ");

scanf("%d", &num); //숫자 입력 받음

n1 = num / 10000; //10000의 자리 계산

n2 = (num - (n1\*10000)) / 1000; //1000의 자리 계산

n3 = (num - (n1\*10000) -(n2\*1000)) / 100; //100의 자리 계산

n4 = (num - (n1\*10000) -(n2\*1000)-(n3\*100)) / 10; //10의 자리 계산

n5 = num % 10; //1의 자리 계산

printf("%d %d %d %d %d \n", n1, n2, n3, n4, n5); //이후 출력

printf("%d %d %d %d \n", n2, n3, n4, n5);

printf("%d %d %d \n", n3, n4, n5);

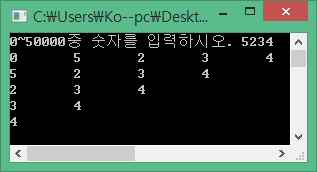
printf("%d %d \n", n4, n5);

printf("%d \n",n5);

return 0; //main함수종료

}

실행결과



Test03

#include <stdio.h> //표준입출력헤더파일 포함

void caln1(int \*n0,int \*n1,int \*n2,int \*n3,int \*n4,int \*n5,int \*n6,int \*n7,int \*n8,int \*n9,int ha)

{ //계산 함수

switch (ha) { //스위치문법을 이용해 0~9중 숫자가 나올 시 값을 증가시킴

case 0:

\*n0=\*n0+1;

break;

case 1:

\*n1=\*n1+1;

break;

case 2:

\*n2=\*n2+1;

break;

case 3:

\*n3=\*n3+1;

break;

case 4:

\*n4=\*n4+1;

break;

case 5:

\*n5=\*n5+1;

break;

case 6:

\*n6=\*n6+1;

break;

case 7:

\*n7=\*n7+1;

break;

case 8:

\*n8=\*n8+1;

break;

case 9:

\*n9+=\*n9+1;

break;

}

}

int main() //main함수

{

int num1, num2, temp, n1, n2, n3, n4; //입력받을 두 변수, 두 숫자중 큰 걸 뒤로 보낼 temp, 자릴 수 마다 분해할 변수생성

int x0 = 0, x1 = 0, x2 = 0, x3 = 0, x4 = 0, x5 = 0, x6 = 0, x7 = 0, x8 = 0, x9 = 0; //0~9의 숫자 증가 변수

printf("첫번째 숫자를 입력해주세요(0~2000):");

scanf("%d", &num1); //첫 번째 숫자 입력받음

printf("두번째 숫자를 입력해주세요(0~2000):");

scanf("%d", &num2); //두 번째 숫자 입력받음

if(num1>num2) //두 숫자 중 작은 숫자를 앞으로 보냄

{

temp = num1;

num1 = num2;

num2 = temp;

}

while(num1<=num2){ //작은 숫자가 큰 숫자가 되면 반복문 종료

n1 = num1 / 1000; //1000의 자리 계산

n2 = (num1 -(n1\*1000)) / 100; //100의 자리 계산

n3 = (num1 -(n1\*1000)-(n2\*100)) / 10; //10의 자리 계산

n4 = num1 % 10; //1의 자리 계산

if(num1>=1000){ //1000의 자리일 경우 n1~n4숫자 전부 유효

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n1);

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n2);

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n3);

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n4);

}

if(num1<1000 && num1>=100){ //100의 자리일 경우 n2~n4숫자만 유효

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n2);

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n3);

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n4);

}

if(num1<100 && num1>=10){ //10의 자리일 경우 n3~n4숫자만 유효

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n3);

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n4);

}

if(num1<10){ //1의 자리일 경우 n4숫자만 유효

caln1(&x0, &x1, &x2, &x3, &x4, &x5, &x6, &x7, &x8, &x9, n4);

}

num1++; //반복문 탈출 조건

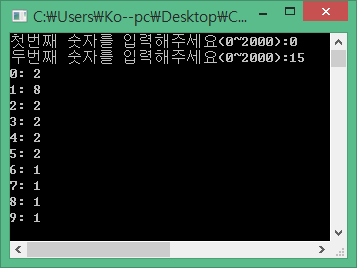
}

printf("0: %d \n1: %d \n2: %d \n3: %d \n4: %d \n5: %d \n6: %d \n7: %d \n8: %d \n9: %d \n",x0, x1, x2, x3, x4, x5, x6, x7, x8, x9); //결과 출력

return 0;

}

실행결과



Test04

#include <stdio.h> //표준입출력헤더파일 포함

#include <stdlib.h> //표준라이브러리헤더파일 포함

int main() //main함수

{

// 1~30의 숫자를 랜덤하게 가지는 2차원 배열

int data[4][4]

= { rand()%31 , rand()%31 ,rand()%31 ,rand()%31 ,

rand()%31 , rand()%31 , rand()%30, rand()%31 ,

rand()%31 , rand()%31 , rand()%31, rand()%31 ,

rand()%31 , rand()%31 , rand()%31, rand()%31 };

int data2[16] ={ //2차원 배열을 버블정렬을 사용하기 위해 1차원 배열로 변환

data[0][0],data[0][1],data[0][2],data[0][3],data[1][0],data[1][1],data[1][2],data[1][3],data[2][0],data[2][1],data[2][2],data[2][3],data[3][0],data[3][1],data[3][2],data[3][3]};

int i,j,temp, result;

printf("랜덤하게 섞여있는 데이터 값\n"); //기존의 섞여있는 값 출력

for (int i = 0; i < 4; i++)

{

for(int j = 0; j < 4; j++)

{

printf("%d ", data[i][j]);

}

printf("\n");

}

for (i = 0; i < 16; i++) //data2값 버블 정렬

for (j = 0; j < 15; j++)

if (data2[j] > data2[j + 1]) {

temp = data2[j];

data2[j] = data2[j + 1];

data2[j + 1] = temp;

}

printf("1차원 배열로 받은 후 오름차순 정렬 \n");

for (int i = 0; i < 16; i++)

{

{

printf("%d ", data2[i]); //정렬한 데이터 출력

}

}

printf("\n");

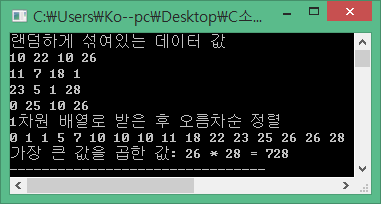
result = data2[15]\*data2[14]; //곱한 값 변수

printf("가장 큰 값을 곱한 값: %d \* %d = %d", data2[14], data2[15], result); //결과 출력

return 0; //main함수 종료

}

실행결과



Test05