|  |
| --- |
| Programming Report 1 |
| 2019-2학기 프로그래밍 04분반 |

|  |
| --- |
| Nathan Cho  jona527111@gmail.com |

# 코드 실행 환경

Visual Studio 2019, Windows 10

운영체제 언어 (Locale) 가 영어로 설정되어 있어 콘솔에 한글 표시가 되지 않아 영어로 작성한 점 양해 부탁드립니다. 모든 출력문 위에 주석으로 한국어 출력을 작성하였습니다.

# 도형의 면적을 계산하는 프로그램을 쓰시오. 도형의 종류는 삼각형, 정사각형, 사각형, 원의 4가지로 한다.

## 문제 해결 방법

먼저 면적을 구할 도형 번호를 입력 받은 후에 각 면적에 따른 입력을 받고 계산했습니다.

유효한 입력 값만 처리해야 되기 때문에 도형 번호를 입력 받는 scanf 함수를 while 반복문을 이용해 유효한 값을 입력 받을 때까지 반복했습니다.

|  |
| --- |
| // printf("\n면적을 구할 도형 번호를 선택하세요. >> ");  printf("\nInput type to calculate area. >> ");  while (1) {  if (scanf("%d", &type) < 1) {  // scanf 버퍼를 확실히 비울 수 있는 방법이 없어서 프로그램의 재시작이 필요함  // 현재 문자를 입력하면 scanf 버퍼가 클리어되지 않고 계속 입력되어 무한 루프가 발생함  // fgets 를 이용해서 한 줄 전체를 입력받고 그 후에 처리해야 함  // printf("오류: 입력은 1~4 사이의 숫자여야 합니다. 프로그램을 재시작하여 주세요.\n");  printf("Error: Input type should be a number within 1 ~ 4.  Please restart the program.\n");  return -1;  }  ...  } |

주석에 작성했듯이 정수를 입력 받는 scanf 함수에 문자를 입력하면 입력 버퍼에 오류가 생겨 무한 반복하는 문제가 있었습니다. scanf 함수 대신 fgets 처럼 오류 처리에 안정적인 함수를 사용하면 문제를 해결할 수 있지만 코드가 복잡해지기 때문에 프로그램을 종료하는 방법을 선택했습니다.

|  |
| --- |
| while (1) {  ...  if (type < 1 || type > 4) {  // printf("오류: 입력은 1~4 사이의 숫자여야 합니다.\n");  // printf("면적을 구할 도형 번호를 선택하세요. >> ");  printf("Error: Input type should be a number within 1 ~ 4.\n");  printf("\nInput type to calculate area. >> ");  }  else {  break;  }  } |

입력이 1~4 사이의 값이 아닐 때는 오류 메시지를 표시하고 다시 입력을 받고 유효한 값이 입력되었을 때는 break 문을 통해 반복문을 빠져나왔습니다.

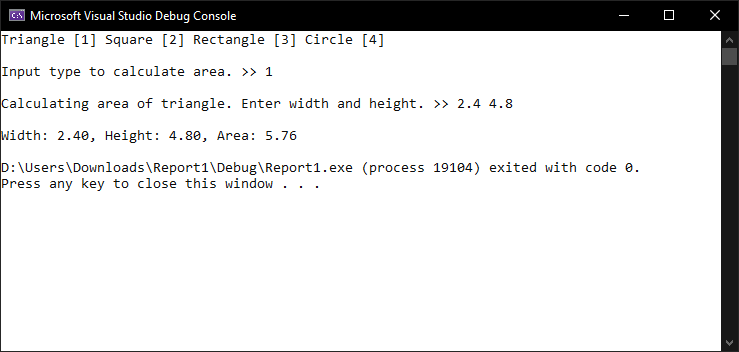
|  |
| --- |
| float width, height;  switch (type) {  case 1:  while (1) {  // printf("\n삼각형의 면적을 구합니다.");  // printf("\n너비와 높이를 입력하세요.");  printf("\nCalculating area of triangle.");  printf(" Enter width and height. >> ");  scanf("%f %f", &width, &height);  if (width < 0 || height < 0) {  // printf("오류: 너비와 높이는 음수가 아니어야 합니다.\n");  printf("Error: Width and height should be non-negative values.\n");  }  else {  break;  }  }  // printf("\n너비: %.2f, 높이: %.2f, 넓이: %.2f\n", width, height, width \* height \* 0.5);  printf("\nWidth: %.2f, Height: %.2f, Area: %.2f\n", width, height, width \* height \* 0.5);  break;  case 2:  ...  break;  case 3:  ...  break;  case 4:  ...  break;  } |

4가지의 도형을 숫자 1~4 로 나누어 type 변수에 저장하고, switch 문을 사용하여 각 도형에 맞는 입력과 계산, 출력을 하도록 제작하였습니다. 또한, 도형의 너비와 높이 값도 유효한 입력값만 처리하도록 scanf와 while반복문을 사용하였습니다. 그리고 도형의 너비, 높이 값이 음수가 되어 넓이가 음수 값이 되지 않도록 예외 처리를 하였습니다.

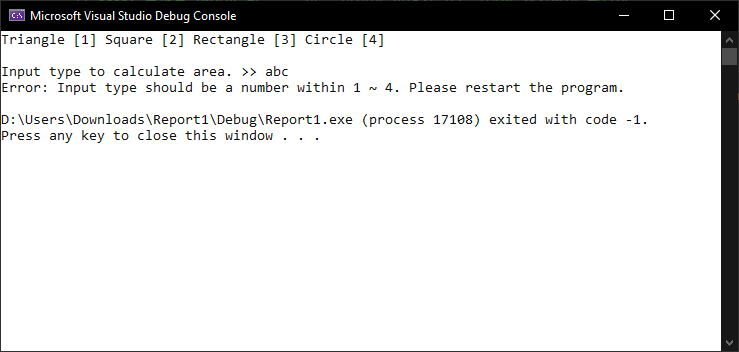
|  |
| --- |
| #define PI 3.141592  int main() {  ...  switch (type) {  ...  case 4:  while (1) {  // printf("\n원의 면적을 구합니다.");  // printf("\n반지름을 입력하세요.");  printf("\nCalculating area of circle.");  printf(" Enter radius. >> ");  scanf("%f", &width);  if (width < 0) {  // printf("오류: 반지름은 음수가 아니어야 합니다.\n");  printf("Error: Radius should be non-negative values.\n");  }  else {  break;  }  }  // printf("\n반지름: %.2f, 넓이: %.2f\n", width, width \* width \* PI);  printf("\nRadius: %.2f, Area: %.2f\n", width, width \* width \* PI);  break;  }  ...  } |

원의 넓이를 구하는 경우에는 반지름은 정의된 width 변수로 사용하고, 원주율 값을math.h 라이브러리를 가져오지 않고 매크로 상수로 정의하여 계산하였습니다.

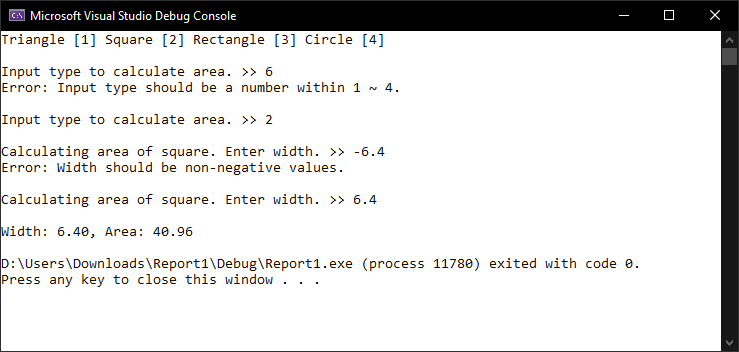
## 수행 결과 예시



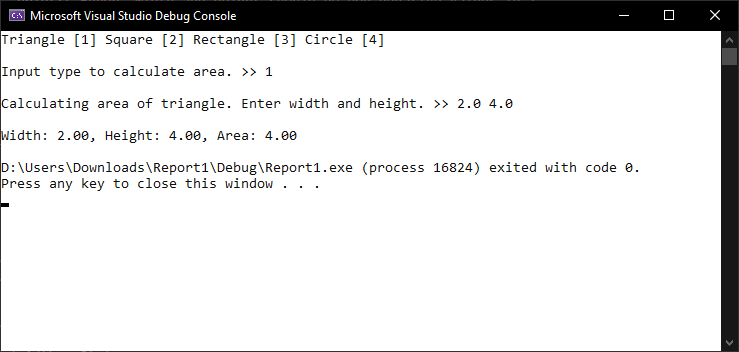
정상 실행 결과



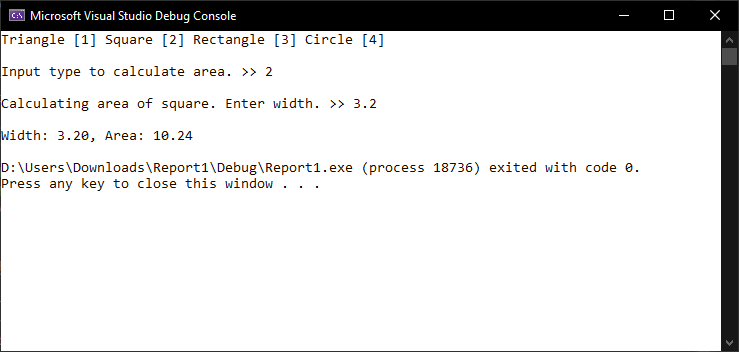
문자 입력 시 오류



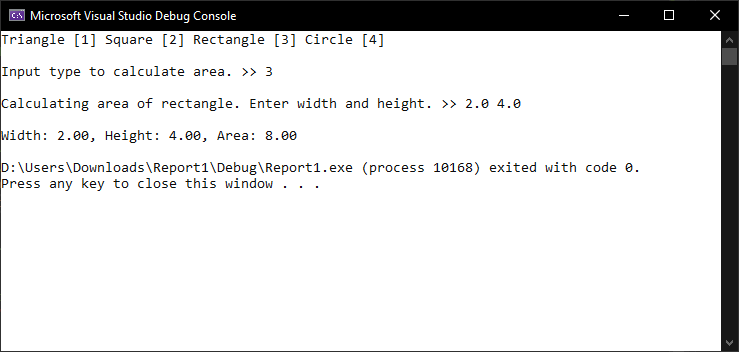
도형 번호 오류, 음수 값 예외 처리



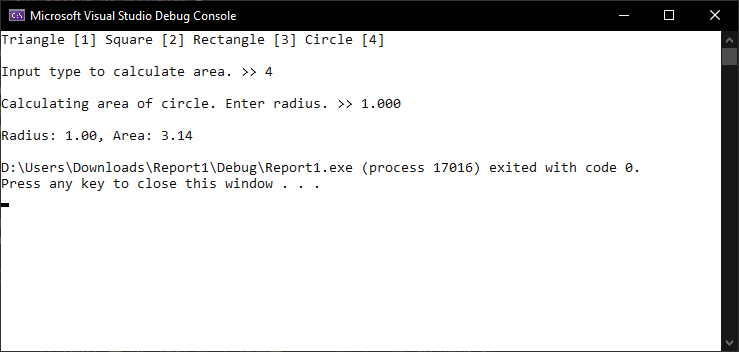
[1] 삼각형의 넓이 계산



[2] 정사각형의 넓이 계산



[3] 직사각형의 넓이 계산



[4] 원의 넓이 계산

# 다음을 참고로 표준입력으로 받은 성별, 신장(키), 몸무게를 이용하여 비만도를 출력하는 프로그램을 작성하시오. 유효한 입력 값이 아니면 다시 입력 받도록 처리해야 한다.

## 문제 해결 방법

1번 문제와 유사하게 2 종류의 입력을 유효한 입력만 처리하는 문제였습니다. 문제에서 입력값의 범위 (200cm 이하, 100kg 이하) 가 제한되어 있고, 비만도의 범위에 따른 출력값이 달라야 한 것이 요점이었습니다.

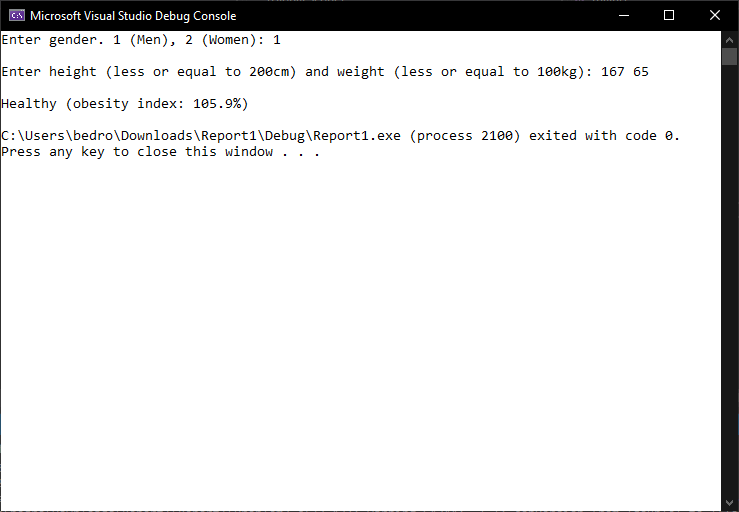
|  |
| --- |
| int gender, height, weight;  ...  // printf("성별을 입력하세요. 1 (남자), 2 (여자):");  printf("Enter gender. 1 (Men), 2 (Women): ");  while (1) {  if (scanf("%d", &gender) < 1) {  // printf("오류: 입력은 1~2 사이의 숫자여야 합니다. ...\n");  printf("Error: ...")  return -1;  }  if (gender < 1 || gender > 4) { ... }  else {  break;  }  }  // printf("\n신장 (200cm 이하) 과 몸무게 (100kg 이하) 를 입력하세요. :");  printf("\nEnter height (less or equal to 200cm) and weight (less or equal to 100kg): ");  while (1) {  if (scanf("%d %d", &height, &weight) < 1) {  ...  return -1;  }  if (height < 0 || height > 200) { ... }  if (weight < 0 || weight > 100) { ... }  if (height >= 0 && height <= 200 && weight >= 0 && weight <= 100) {  break;  }  else {  // printf("오류: 몸무게는 0~100 (kg) 사이의 숫자이고  신장은 0~200 (cm) 사이의 숫자여야 합니다.\n");  // printf("\n신장 (200cm 이하) 과 몸무게 (100kg 이하) 를 입력하세요. :");  printf("Error: ...  printf("\nEnter ...");  }  } |

1번 문제의 반복문을 다시 이용하여 유효한 입력값을 받았습니다. 신장과 몸무게의 허용 범위에 맞는 예외 처리 또한 추가하였습니다.

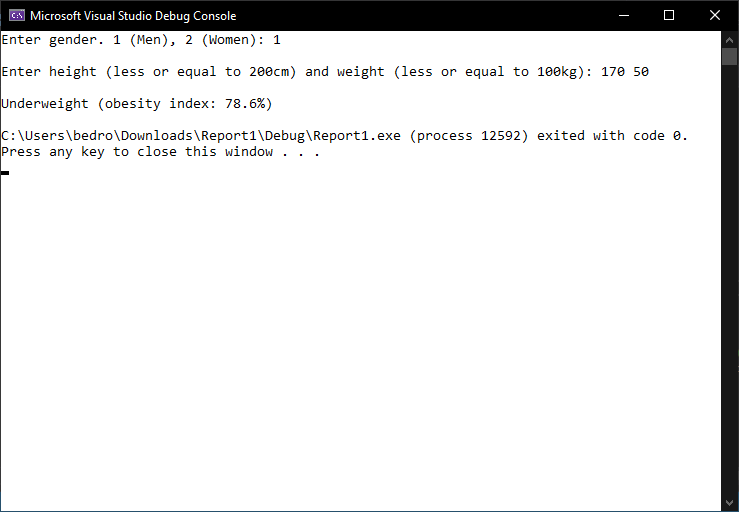
|  |
| --- |
| if (gender == 1) {  average = height \* height \* 22 \* 0.0001;  }  else if (gender == 2) {  average = height \* height \* 21 \* 0.0001;  }  else {  // printf("오류: ...\n");  ...  return -1;  }  obesity = weight / average;  printf("\n");  if (obesity < 0.9) {  // printf("저체중 (비만도: %.1f\%)", obesity \* 100.0);  printf("Underweight (obesity index: %.1f%%)", obesity \* 100.0);  }  else if (obesity >= 0.9 && obesity <= 1.1) {  // printf("정상 (비만도: %.1f\%)", obesity \* 100.0);  printf("Healthy (obesity index: %.1f%%)", obesity \* 100.0);  }  else if (obesity > 1.1 && obesity <= 1.2) {  // printf("과체중 (비만도: %.1f\%)", obesity \* 100.0);  printf("Overweight (obesity index: %.1f%%)", obesity \* 100.0);  }  else if (obesity > 1.2) {  // printf("비만 (비만도: %.1f\%)", obesity \* 100.0);  printf("Obese (obesity index: %.1f%%)", obesity \* 100.0);  } |

성별에 따른 비만도를 계산하고 비만도의 범위에 따라 저체중, 정상, 과체중, 비만을 출력하였습니다. 비만도의 범위는 if, else if, else 문을 사용하여 분류하였습니다.

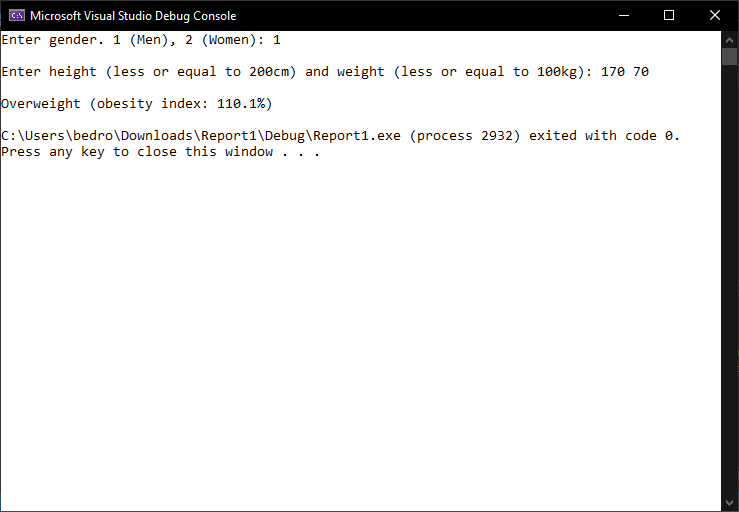
## 수행 결과 예시



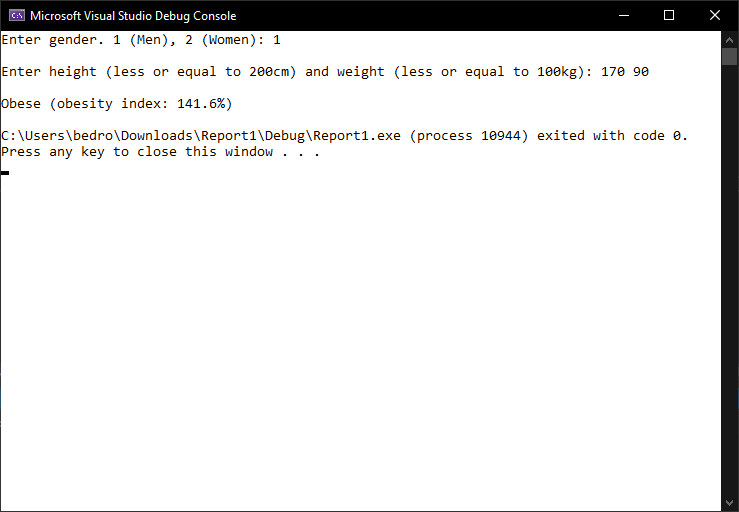
정상 실행 결과 (비만도: 정상)



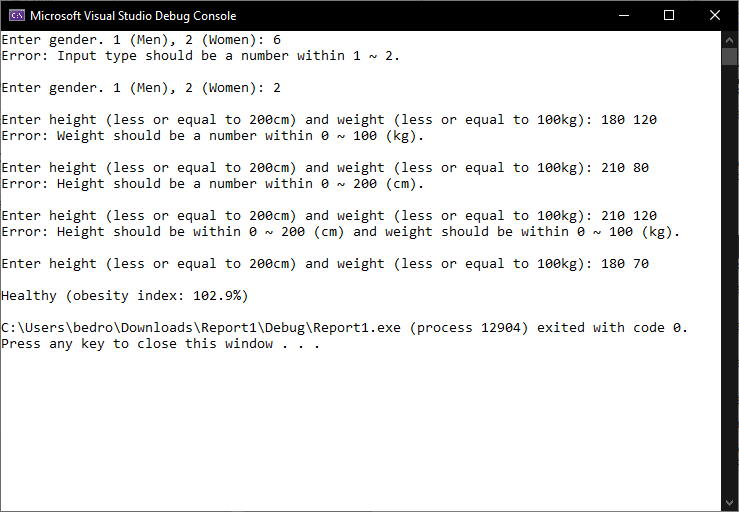
정상 실행 결과 (비만도: 저체중)



정상 실행 결과 (비만도: 과체중)



정상 실행 결과 (비만도: 비만)



입력값의 범위 초과 예외 처리

# 표준입력으로 입력한 정수에서 각각의 자리에 해당하는 수를 반대로 출력하는 프로그램을 do while 문을 이용하여 작성하시오. 정수입력이 아니면 다시 입력 받도록 처리해야 한다.

## 문제 해결 방법

입력 받은 정수를 거꾸로 출력하는 가장 쉬운 방법은 입력을 문자열로 받고 역순으로 출력하는 것입니다. 하지만 정수를 입력 받는 음수도 처리해야 하기 때문에 scanf 에서 %d를 이용하여 입력 받는 방법을 선택했습니다.

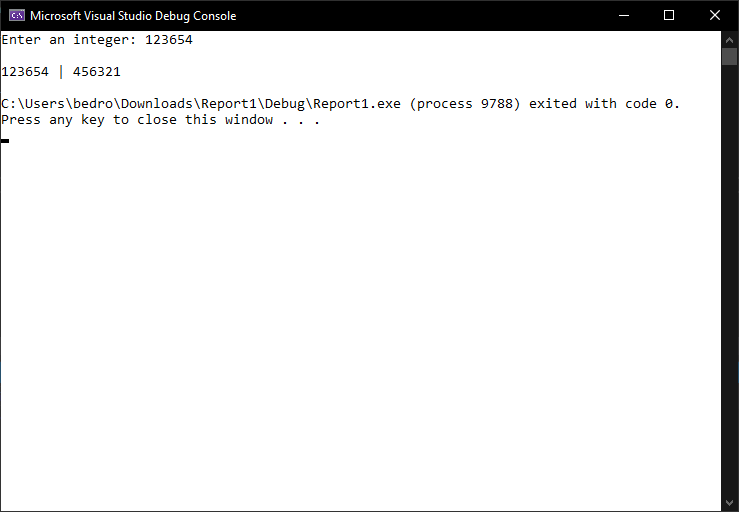
|  |
| --- |
| int input, success;  // printf("정수를 입력하세요. :");  printf("Enter an integer: ");  do {  success = scanf("%d", &input);  if (success == 0) {  // printf("오류: 입력은 정수여야 합니다.\n");  printf("Error: Input type should be an integer.\n\n");  // 입력 버퍼 클리어  scanf("%\*[^\n]");  // printf("정수를 입력하세요. :");  printf("Enter an integer: ");  }  } while (!success); |

3번 문제에서는 1, 2번 문제와 달리 유효한 입력만을 처리하는 방법을 달리해 프로그램을 종료해야만 했던 문제를 해결하였습니다. scanf("%\*[^\n]"); 를 사용해 scanf 함수에 남아있는 버퍼를 비울 수 있도록 하여 정수가 아닌 문자가 입력되더라도 다시 while 문으로 입력 받을 수 있도록 하였습니다.

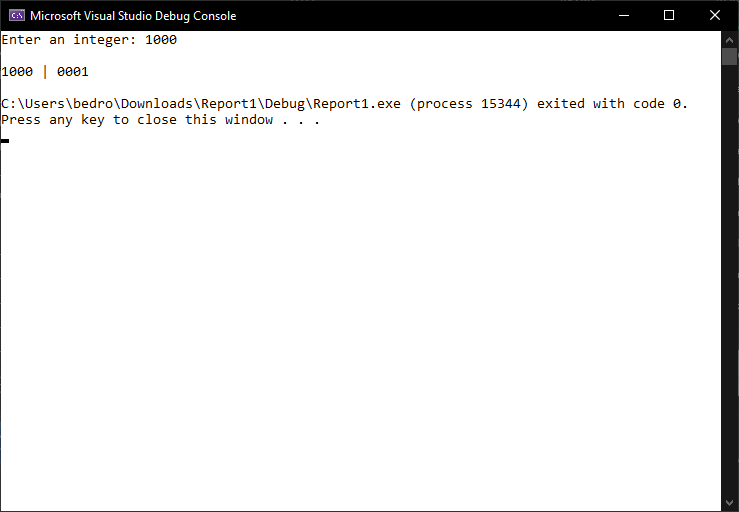
|  |
| --- |
| int count = 0;  int current;  if (input == 0) {  printf("0");  }  else {  if (input < 0) {  // 음수이면 부호를 앞에 출력  printf("-");  // 양수로 변환  input \*= -1;  }  while (1) {  current = input;  for (int i = 0; i < count; i++) {  // 현재 자리의 값 계산  current /= 10;  }  if (current == 0) {  break;  }  else {  // 값을 10으로 나눈 나머지를 출력  printf("%d", current % 10);  count++;  }  }  } |

숫자를 거꾸로 출력하는 부분에는 입력 값을 10으로 나눈 몫을 다시 10으로 나눈 나머지를 반복하여 출력하는 방법으로 제작하였습니다. LSB 부터 MSB 까지 반복하면서 출력해 입력을 거꾸로 표시하도록 제작하였습니다. 또한 입력 값이 0 이나 음수일 때의 출력 오류를 방지하는 예외처리를 하였습니다.

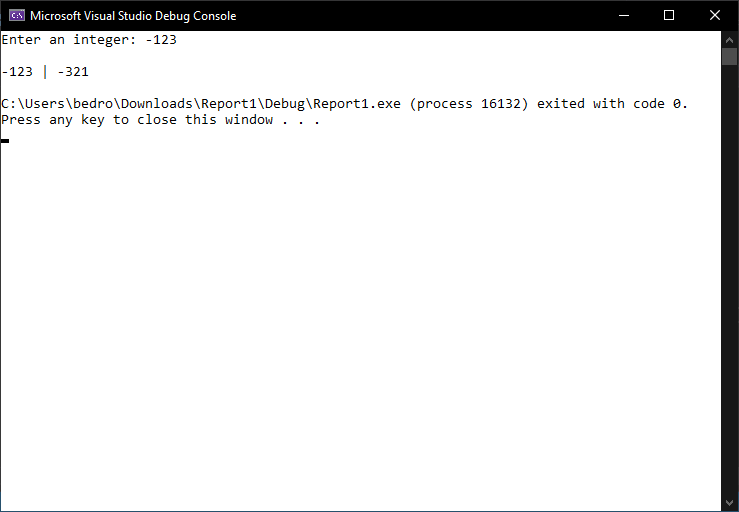
## 수행 결과 예시



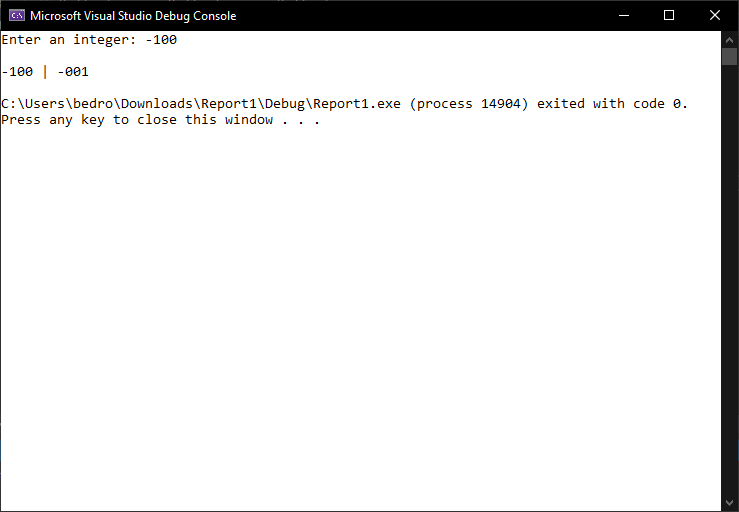
정상 실행 결과



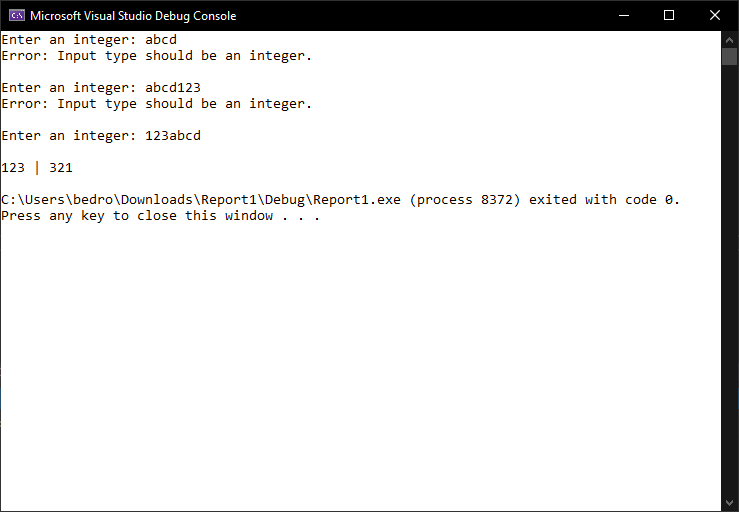
0으로 끝나는 정수 입력



음수 입력



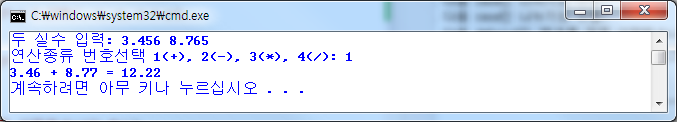
0으로 끝나는 음수 입력



문자열 입력 예외 처리

문자열로 시작하는 입력은 예외 처리를 할 수 있지만 숫자로 시작하는 입력은 scanf 함수에서 예외 처리가 불가능합니다.

# 다음을 참고로 표준입력으로 받은 두 실수의 연산을 수행하는 프로그램을 작성하시오. switch 문을 사용하고, 다음과 같은 입출력을 참고하여 아래 결과에서 연산 종류를 번호선택으로 하는 대신 문자선택(+, -, \*, /)으로 하도록 프로그램을 수정하시오.



## 문제 해결 방법

2번 문제와 유사하게 두 가지 종류의 입력을 받는 문제입니다. 하지만 두 번째 입력이 문자인 것이 요점이었습니다. 계산할 값들을 입력 받고 연산자를 입력 받아 switch 문을 사용해서 연산자 별로 계산하였습니다.

|  |
| --- |
| float a, b;  char operand;  int success;  // printf("2개의 실수를 입력하세요. :");  printf("Enter 2 real numbers: ");  do {  success = scanf("%f %f", &a, &b);  if (success == 0) {  // printf("오류: 입력은 2개의 실수여야 합니다.\n");  printf("Error: Input type should be 2 real numbers.\n\n");  // 입력 버퍼 클리어  scanf("%\*[^\n]");  // printf("2개의 실수를 입력하세요. :");  printf("Enter 2 real numbers: ");  }  } while (!success);  // printf("연산 종류를 입력하세요. :");  printf("Enter operand (+ - \* /): ");  do {  success = scanf(" %c", &operand);  if (success == 0) {  ...  }  } while (!success); |

3번 문제에서 사용했던 정수 입력 예외 처리를 재사용하여 실수 입력 예외 처리를 하였습니다. 하지만 두 번째 입력, scanf(" %c", &operand); 부분에서 두 실수를 입력하고 마지막에 입력하는 "\n" 문자가 넘어와 입력을 받지 않고 무한 반복을 하는 문제가 있었습니다. 그 문제는 포맷 문자열 "%c" 앞에 공백 문자를 하나 입력해 해결했습니다. scanf 함수가 포맷 문자열에 공백 문자가 있으면 띄어쓰기나 엔터 같은 공백 문자를 무시하는 특징을 사용했습니다.

|  |
| --- |
| // printf("연산 종류를 입력하세요. :");  printf("Enter operand (+ - \* /): ");  do {  success = scanf(" %c", &operand);  if (operand != '+' && operand != '-' && operand != '\*' && operand != '/') {  success = 0;  }  if (success == 0) {  ...  }  } while (!success); |

연산자를 입력 받는 부분에도 4가지 연산자만 허용되도록 예외 처리를 하였습니다.

|  |
| --- |
| printf("%.2f %c %.2f = ", a, operand, b);  switch (operand) {  case '+':  printf("%.2f", a + b);  break;  case '-':  printf("%.2f", a - b);  break;  case '\*':  printf("%.2f", a \* b);  break;  case '/':  printf("%.2f", a / b);  break;  } |

실제로 입력 받은 문자열을 switch 문으로 case를 나누어 입력한 실수에 대한 계산을 하였습니다.

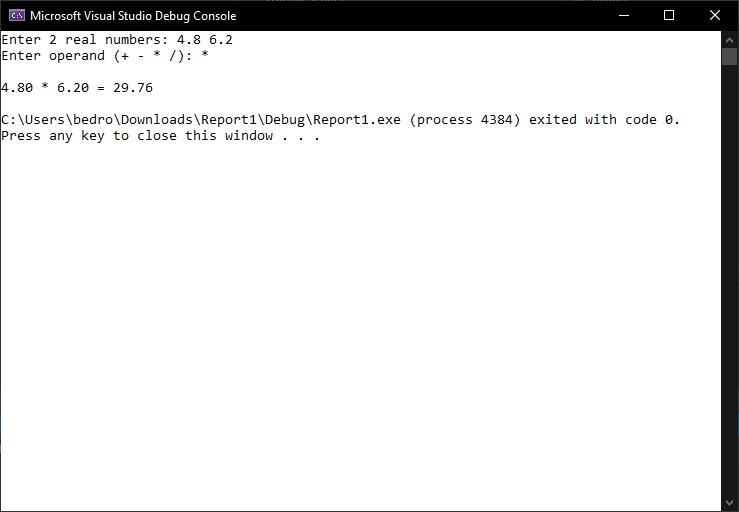
## 수행 결과 예시



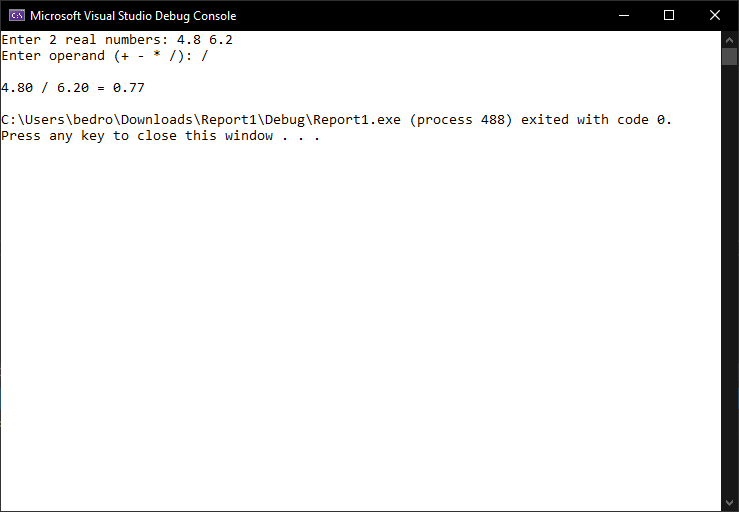
정상 실행 결과 (덧셈)



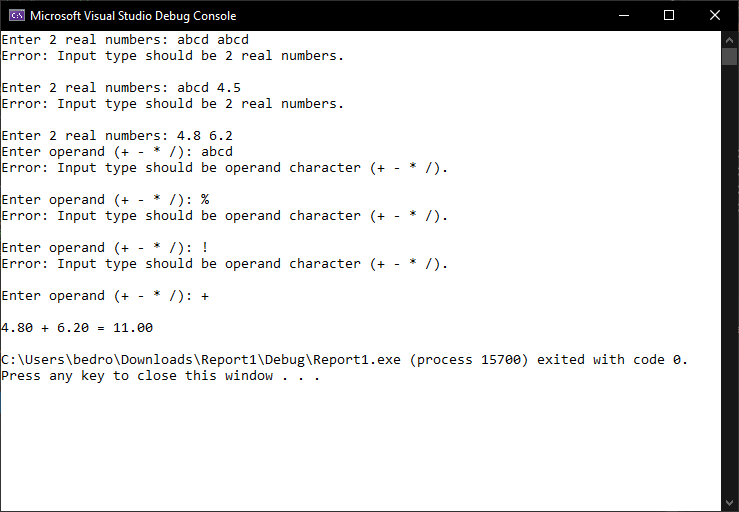
정상 실행 결과 (뺄셈)



정상 실행 결과 (곱셈)



정상 실행 결과 (나눗셈)



실수 입력, 연산자 문자 입력 예외 처리

문자열로 시작하는 입력은 예외 처리를 할 수 있지만 숫자로 시작하는 입력은 scanf 함수에서 예외 처리가 불가능합니다.