

CSED101. Programming & Problem solving

Spring 2021

Programming Assignment #1 (75 points)

김재우 (zkim08@postech.ac.kr)

■ 제출 마감일: 2021.04.10 23:59

■ 개발 환경: Windows Visual Studio 2019

■ 제출물

- C 소스 코드 (assn1.c)
 - 프로그램의 소스 코드에 채점자의 이해를 돕기 위한 주석을 반드시 붙여주세요.
- 보고서 파일 (.docx, .hwp 또는 .pdf; assn1.docx, assn1.hwp 또는 assn1.pdf)
 - 보고서는 AssnReadMe.pdf를 참조하여 작성하시면 됩니다.
 - 명예 서약 (Honor code): 표지에 다음의 서약을 기입하여 제출해 주세요: “나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.” 보고서 표지에 명예 서약이 기입되어 있지 않은 과제는 제출되지 않은 것으로 처리됩니다.
 - 작성한 소스 코드와 보고서 파일은 PLMS를 통해 제출해 주세요.

■ 주의 사항

- 컴파일이나 실행이 되지 않는 과제는 0점으로 채점됩니다.
- 제출 기한보다 하루 늦게 제출된 과제는 최종 20%, 이를 늦게 제출된 과제는 최종 40% 감점됩니다. 제출 기한보다 사흘 이상 늦으면 제출 받지 않습니다 (0점 처리).
- 각 문제의 제한 조건과 요구 사항을 반드시 지켜 주시기 바랍니다.
- 모든 문제의 출력 형식은 채점을 위해 아래에 제시된 예시들과 최대한 비슷하게 작성해 주세요.
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 “POSTECH 전자컴퓨터공학부 부정행위 정의”를 따릅니다 (PLMS의 본 과목 공지사항에 등록된 글 중, 제목이 [document about cheating]인 글에 첨부되어 있는 disciplinary.pdf를 참조하세요).
- 이번 과제는 추가 기능 구현과 관련된 추가 점수가 따로 없습니다.

■ Problem

(문제)

이 문제는 크게 3개의 파트로 나뉘져 있습니다. 이 3가지 문제를 조건 내에 주어진 input을 통해서 결과값을 내야합니다.

1. Riemann sum estimation
2. Printing out Roman numerals
3. Mystery number calculation

(목적)

이번 과제를 통하여 조건문, 반복문, 사용자 정의 함수 및 라이브러리 함수 사용법을 익힙니다.

(주의사항)

1. 이번 과제는 함수를 정의하고 사용하는 방법을 익히는 문제이므로 main() 함수에 모든 기능을 구현한 경우 감점 처리 합니다.
2. 문서에 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인 후 구현하도록 합니다. 이 때, 설명에서 지정한 사용자 정의 함수의 매개변수의 개수와 자료형, 함수 이름, 반환 자료형 등은 자유롭게 변경이 가능합니다. 그러나 동일한 기능을 하는 함수는 반드시 있어야 하며, 변경 시 무엇을 어떻게 변경해서 구현했는지 보고서에 기록하도록 합니다. 이외에 필요한 함수는 정의해서 사용할 수 있습니다.
3. 프로그램 구현 시, main() 함수를 호출을 직접 하지 않습니다. 즉, 소스 코드 내에 main(); 이라고 호출하지 않습니다.
4. 전역 변수, 배열 및 goto 문은 사용할 수 없으며, 포인터의 경우 수업시간에 다룬 내용에 한해서 사용이 가능합니다.
5. 사용자 입력에서 숫자를 입력 받는 부분에는 숫자만 입력하는 것으로 가정합니다. 즉, 숫자 입력 받는 부분에는 문자 등의 입력에 대해서는 고려할 필요가 없습니다.
6. 명시된 예러 처리 외에는 고려하지 않아도 됩니다.
7. 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.

I. 초기 선택 메뉴 (5점)

- (1) 아래와 같이 3가지 기능을 실행할 수 있는 프로그램을 작성하여 사용자로부터 원하는 기능을 선택할 수 있도록 합니다.

```
=====
CSED 101 Assignment 1
Choose from the following:

1. Riemann Sum
2. Roman Numerals
3. The Mystery Number
4. Terminate Program
=====
Selection:
```

- (2) 1, 2, 3, 4 외의 다른 숫자가 입력되면 에러 메시지를 출력하고, 번호를 다시 입력 받습니다.
1~4에 해당되는 숫자를 입력 받을 때까지 다시 입력 받으며, 해당되는 숫자를 입력한 경우 해당 기능을 수행하게 됩니다.
- (3) 사용자가 4를 입력한 경우에는 프로그램을 종료합니다.
- (4) 이 문제와 관련하여 아래 사용자 정의 함수를 반드시 구현 후, 사용해야 합니다.
- **int inputSelection():** 메뉴 화면을 출력하고, 사용자로부터 메뉴 선택을 입력 받아서 반환

아래 실행 예시의 빨간색 밑줄은 사용자 입력에 해당 합니다.

```
=====
CSED 101 Assignment 1
Choose from the following:

1. Riemann Sum
2. Roman Numerals
3. The Mystery Number
4. Terminate Program
=====
Selection: -1
Invalid selection

Selection: 124
Invalid selection

Selection: 4

Program Terminated.
계속하려면 아무 키나 누르십시오 . . .
```

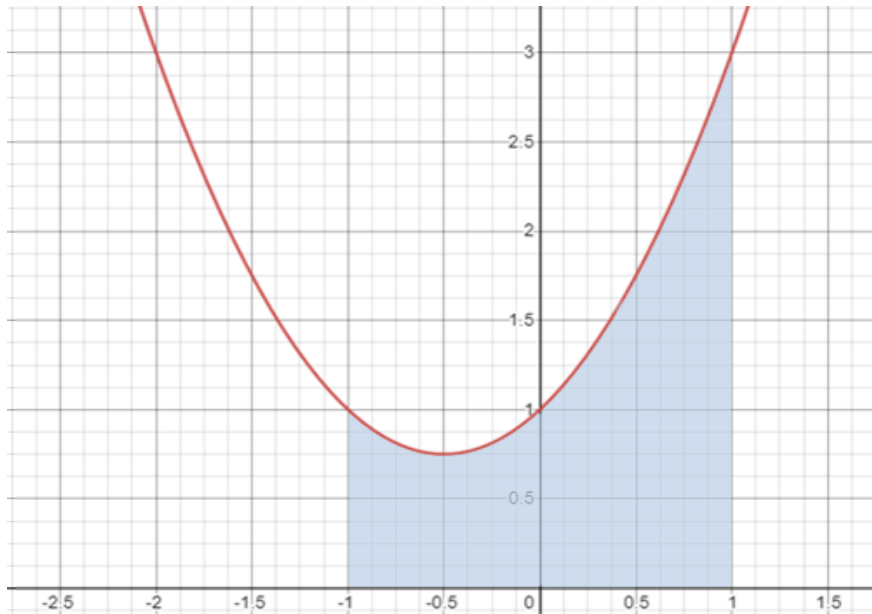
II. Riemann Sum (30점)

(문제)

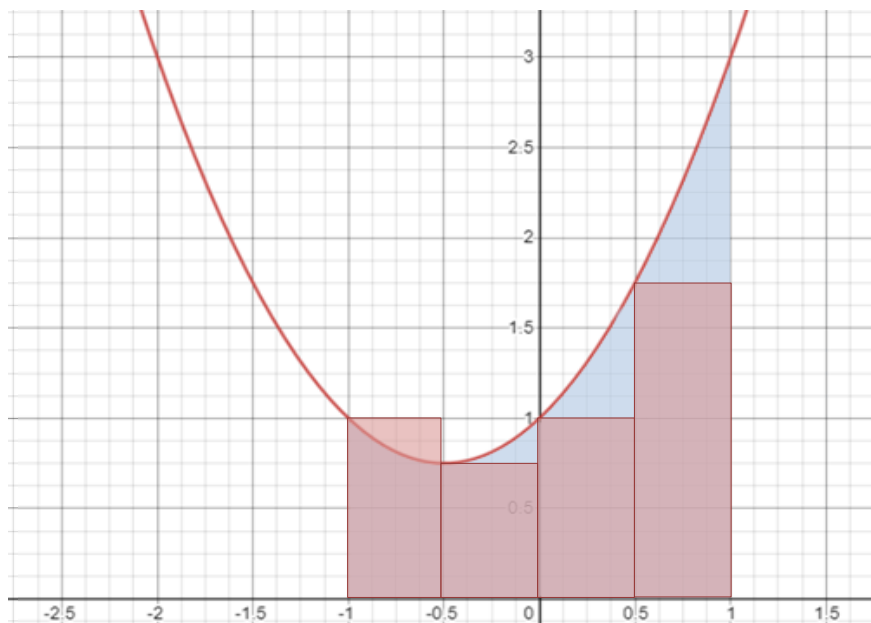
Riemann sum (리만 합)은 어떤 곡선의 면적을 사각형으로 근사값을 내는 겁니다. 이 문제에서는 2차 함수 ($ax^2 + bx + c$) 의 곡선 아래의 면적을 리만 합으로 근사값을 구합니다. 그리고 리만 합에 사용할 사각형의 수, x 의 범위, 그리고 리만 합의 종류(왼쪽 리만 합, 가운데 리만 합, 오른쪽 리만 합)를 사용자로부터 입력 받아 추정값을 계산합니다.

(설명)

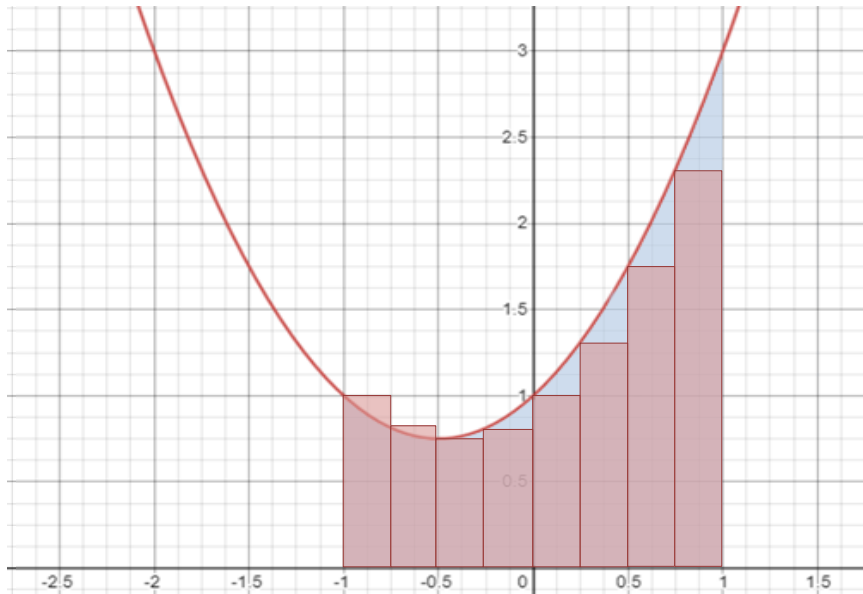
아래 곡선 아래의 면적(하늘색)을 구한다고 생각해 봅시다.



정확한 면적을 구하기는 어렵지만, 사각형을 사용해 근사값을 구할 수 있습니다.



사각형의 수가 많을수록, 즉 사각형의 폭이 좁고 촘촘하게 들어간다면 정밀도가 높아져 실제의 면적과 더욱 비슷하게 나옵니다.



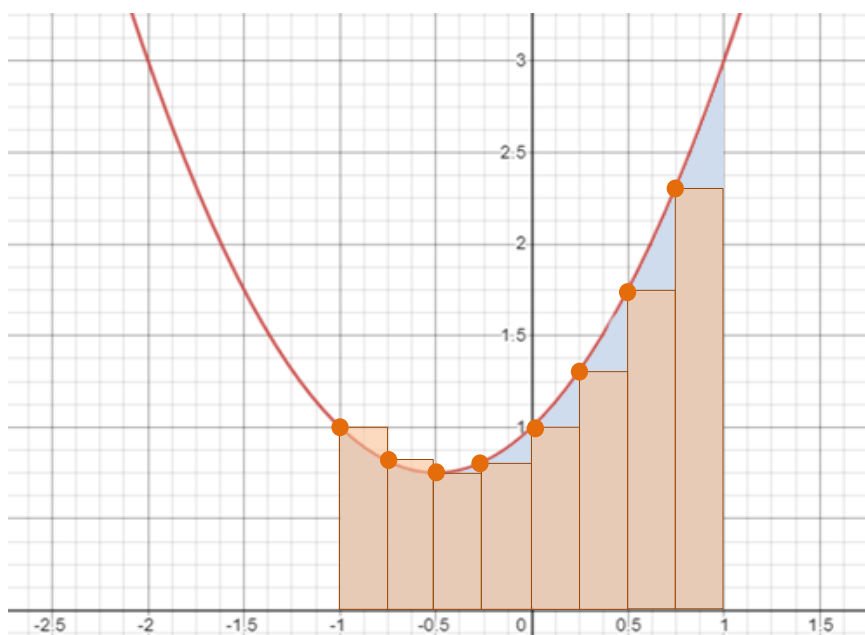
이러한 종류의 근사를 리만 합이라고 하며, 리만 합은 아래 3가지 종류로 나눌 수 있습니다.

먼저, 아래의 조건에 해당하는 곡선에 대해서 면적을 구하는 예를 들어 봅시다.

- 곡선: $x^2 + x + 1$
- 사각형의 수: 8개
- x 의 범위: $-1 \sim 1$

x 의 범위가 -1 부터 1 이고 사각형의 수는 8이니 사각형 1개의 폭은 (가로 길이) / (사각형의 수)로 계산하여 $2 / 8 = 0.25$ 가 됩니다. 이 값을 dx 라고 하면, 사각형의 각각의 면적은 $dx * (\text{세로 길이})$ 가 됩니다.

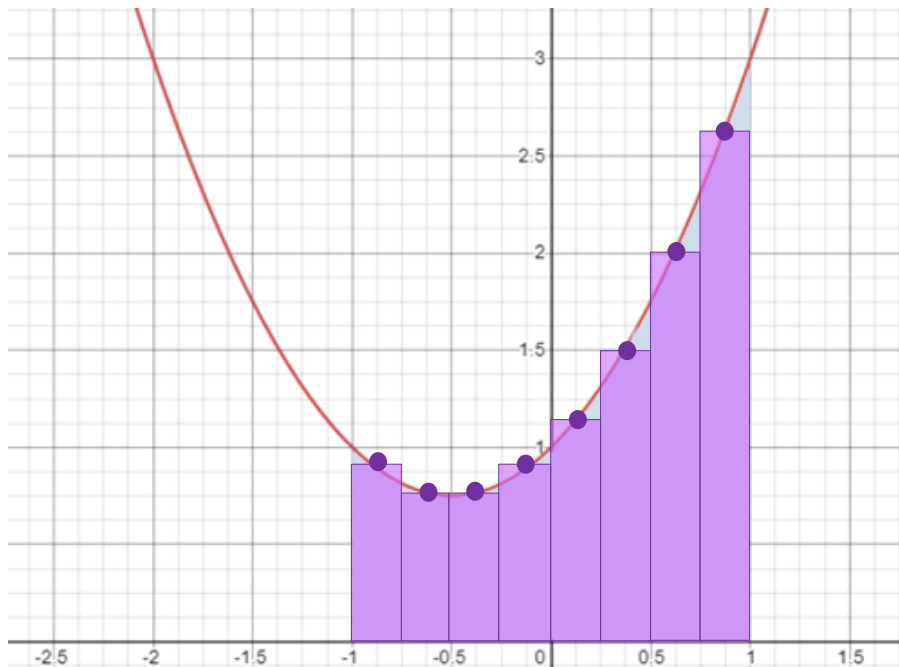
A. 왼쪽 리만 합 (Left hand sum)



왼쪽 리만 합은 사각형의 세로 길이가 dx 의 왼쪽 부분의 영향을 받습니다. x 가 dx 의 왼쪽의 끝일 때 나오는 곡선의 값입니다. 그 사각형의 면적을 다 더해서 나오는 값이 리만 합 추정값이 됩니다.

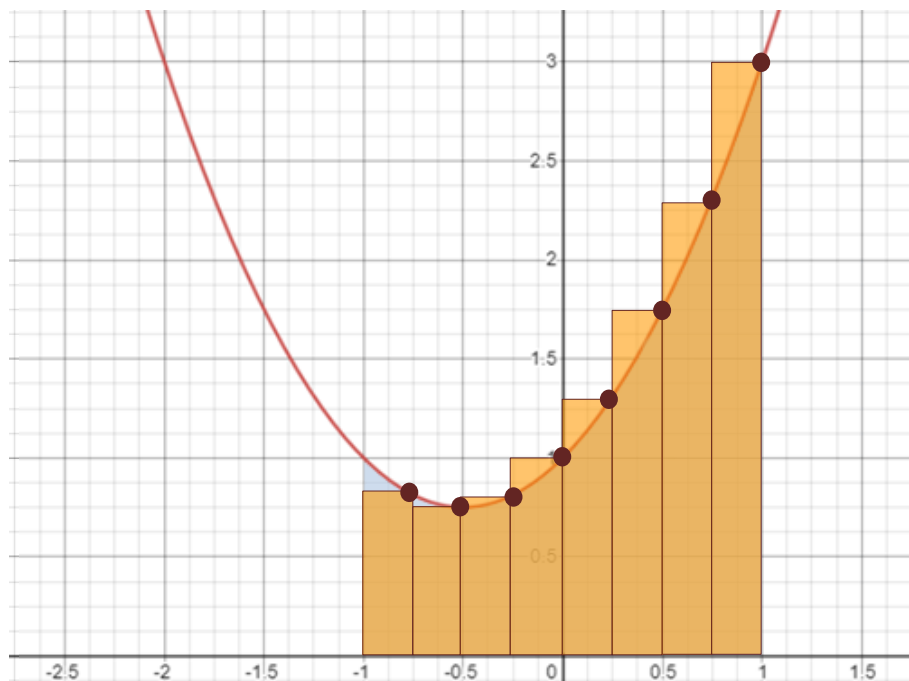
B. 가운데 리만 합 (Midpoint sum)

가운데 리만 합은 사각형의 세로 길이가 x 가 dx 의 중간일 때 나오는 곡선의 값입니다



C. 오른쪽 리만 합 (Right hand sum)

오른쪽 리만 합은 사각형의 세로 길이가 x 가 dx 의 오른쪽 끝일 때 나오는 곡선의 값입니다.



더 자세한 내용은 이 링크를 참고해 주세요: https://en.wikipedia.org/wiki/Riemann_sum

(요구사항)

- (1) 사용자로부터 2차 함수 ($ax^2 + bx + c$)의 계수 a , b , c 를 순서대로 입력 받습니다. **a 가 0이거나 b 가 0의 경우도 계산이 가능해야 합니다.**
과제 채점 시 곡선의 계수들은 정수만 넣을 것이며 사용자가 넣은 계수들이 정수가 맞는지 확인하는 절차는 필요 없습니다. (the coefficients will all be integers and do not require validation)
- (2) 면적을 구할 가로 길이에 해당하는 x 의 범위의 시작(initial value)과 끝(final value) 값을 입력 받습니다. 단, **final value는 initial value 보다 큰 수가 입력되어야 합니다.** x 의 범위는 프로그램 상으로는 아주 큰 범위까지도 계산이 가능하나 이 과제의 경우엔 **± 50 이 넘는 x 의 범위를 테스트하지 않겠습니다.**
과제 채점 시 1번의 계수들과 똑같은 조건으로 정수만 넣을 것입니다.
- (3) 사각형의 수를 입력 받습니다. 단, **양의 정수**를 입력해야 합니다.
- (4) 리만 합의 세 종류를 계산 후에 **세 결과물을 다 printf를 써서 출력**합니다. 각 결과에 대해서 **소수 4자리**까지 출력합니다. 결과물이 어떻게 나와야 하는지는 실행 예시를 참고해 주세요.
- (5) 사용자로부터 리만 합 구하기를 계속할 것인지 여부를 입력 받아 'y'를 입력 받은 경우에는 2차 방정식의 계수부터 다시 입력 받기를 시작하고, 'n'을 입력 받은 경우에는 초기 선택 메뉴로 돌아갑니다. 'y'와 'n' 입력 외에는 고려하지 않습니다.
- (6) 이 문제와 관련하여 **아래 사용자 정의 함수**를 반드시 구현 후, 사용해야 합니다.
 - void funcRiemannSum(): 초기 선택 메뉴에서 사용자가 1을 입력한 경우에 실행되는 함수로, 사용자로부터 방정식의 계수 등을 입력 받아, 리만 합을 계산하여 출력하는 기능을 수행합니다. 실제 이 함수에서 아래 리만 합 계산하는 함수(들) 호출하여 사용합니다.
 - 리만 합 계산하는 함수(들): 크게 2가지 방법이 있는데 하나는 계산방법 3개를 한 함수에 넣는 방법이 있고, 다른 하나는 아래처럼 각각의 계산방법을 3개의 함수로 만드는 것입니다.
 - ✓ **void calcLefthand()**: 왼쪽 리만 합을 계산하고 그 결과를 출력
 - ✓ **void calcRighthand()**: 오른쪽 리만 합을 계산하고 그 결과를 출력
 - ✓ **void calcMidpoint()**: 가운데 리만 합을 계산하고 그 결과를 출력리만 합 계산을 위해, 위 두가지 중 하나를 사용해도 되고 다른 방법으로 구현해도 됩니다. 단, 리만 합 계산하는 기능을 가진 함수가 반드시 있어야 합니다.

예외처리) 사용자 입력에서 적절하지 않은 입력 값이 들어 온 경우, 에러 메시지 출력 후 적절한 범위 내에 있는 입력 값이 들어올 때까지 반복하여 입력을 받도록 합니다. 에러 메시지 출력 부분은 아래의 실행 예시를 참고해서 작성하세요.

(실행 예시 1)

```
=====
CSED 101 Assignment 1
Choose from the following:

1. Riemann Sum
2. Roman Numerals
3. The Mystery Number
4. Terminate Program
=====
Selection: 1
```

```

Enter integer coefficients [a b c]: 1 1 1
Enter the initial value of x: -1
Enter the final value of x: 1

Enter the number of approximating rectangles: 5

Estimated Area:

Lefthand sum: 2.3200 sq. units
Midpoint sum: 2.6400 sq. units
Righthand sum: 3.1200 sq. units

Do you want to continue(y/n)? y

Enter integer coefficients [a b c]: 1 1 1
Enter the initial value of x: -1
Enter the final value of x: -2
Final value of x must be greater than initial value

Enter the final value of x: 1

Enter the number of approximating rectangles: -10
Number of rectangles must be positive

Enter the number of approximating rectangles: -4
Number of rectangles must be positive

Enter the number of approximating rectangles: 10

Estimated Area:

Lefthand sum: 2.4800 sq. units
Midpoint sum: 2.6600 sq. units
Righthand sum: 2.8800 sq. units

Do you want to continue? (y/n) n

=====
CSED 101 Assignment 1
Choose from the following:

1. Riemann Sum
2. Roman Numerals
3. The Mystery Number
4. Terminate Program
=====
Selection:

```


(실행 예시 2)

```
=====
CSED 101 Assignment 1
Choose from the following:

1. Riemann Sum
2. Roman Numerals
3. The Mystery Number
4. Terminate Program
=====
Selection: 1

Enter integer coefficients [a b c]: 0 3 -9
Enter the initial value of x: -1
Enter the final value of x: 1

Enter the number of approximating rectangles: 10

Estimated Area:

Lefthand sum: -18.6000 sq. units
Midpoint sum: -18.0000 sq. units
Righthand sum: -17.4000 sq. units

Do you want to continue? (y/n) y

Enter integer coefficients [a b c]: 3 4 0
Enter the initial value of x: -1
Enter the final value of x: 3

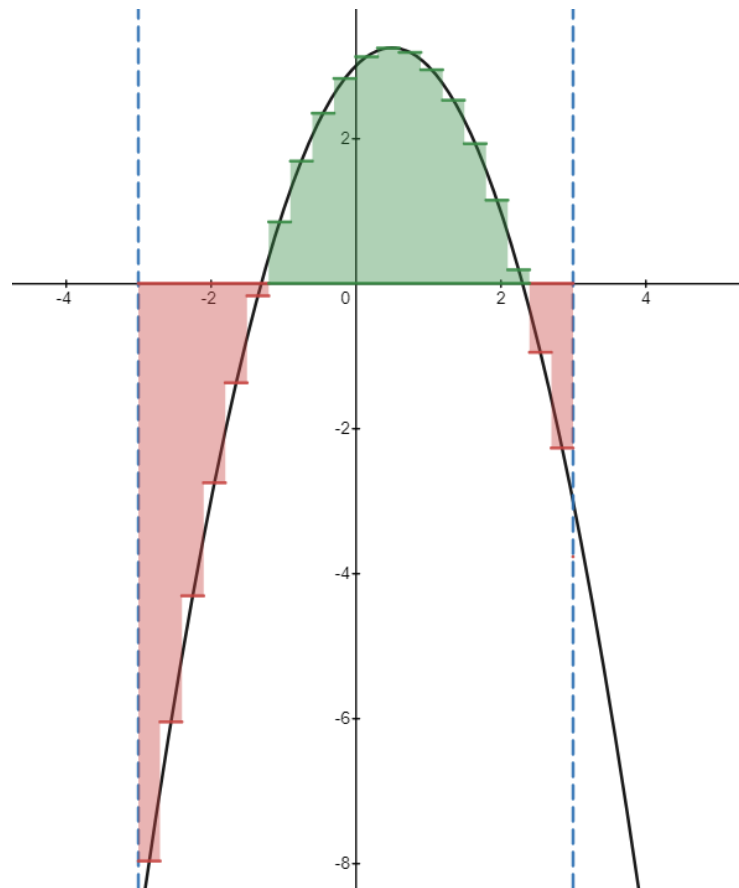
Enter the number of approximating rectangles: 19

Estimated Area:

Lefthand sum: 39.8781 sq. units
Midpoint sum: 43.9557 sq. units
Righthand sum: 48.2992 sq. units

Do you want to continue? (y/n)
```

아래의 예시에 쓰인 곡선은 $-x^2+x+3$ 입니다. x 의 범위가 꽤 커서 곡선이 이러한 면적을 가질 경우엔 빨강색 면적은 음수, 초록색 면적은 양수입니다.



아래 실행 예시 3은 프로그램에 위와 같은 곡선을 입력했을 경우의 예시에 해당하며, 그 결과를 확인 할 수 있습니다.

(실행 예시 3)

```
=====
```

```
CSED 101 Assignment 1
```

```
Choose from the following:
```

1. Riemann Sum
2. Roman Numerals
3. The Mystery Number
4. Terminate Program

```
=====
```

```
Selection: 1
```

```
Enter integer coefficients [a b c]: -1 1 3
```

```
Enter the initial value of x: -3
```

```
Enter the final value of x: 4
```

```
Enter the number of approximating rectangles: 12
```

```
Estimated Area:
```

Lefthand sum: -6.2303 sq. units
Midpoint sum: -5.6348 sq. units
Righthand sum: -6.2303 sq. units

Do you want to continue? (y/n) y

Enter integer coefficients [a b c]: -1 1 3

Enter the initial value of x: -3

Enter the final value of x: 3

Enter the number of approximating rectangles: 20

Estimated Area:

Lefthand sum: -0.9900 sq. units

Midpoint sum: 0.0450 sq. units

Righthand sum: 0.8100 sq. units

Do you want to continue? (y/n)

III. 로마 숫자 변환 (15점)

(문제) 10진수로 입력 받은 정수(범위: 1 ~ 1000까지)를 로마 숫자로 변환하는 프로그램을 작성 하시오.

(설명)

옛날 고대 로마에서는 숫자를 쓸 때 알파벳을 써서 표현했습니다. 현재는 아라비아 숫자(1 2 3 4 등)를 훨씬 흔하게 쓰고 있지만, 로마 숫자도 군주의 호칭 (e.g. Elizabeth II, Louis VII), 올림픽의 공식 명칭 (e.g. Games of the XXXI Olympiad / Rio 2016), 또는 시계 등의 아이템에도 사용되기도 합니다.

로마 숫자는 7개의 기본 기호를 조합하여 수를 나타냅니다. 기본 기호와 값은 아래와 같습니다.

기호	I	V	X	L	C	D	M
값	1	5	10	50	100	500	1000

예를 들면 I(1)가 3개면 III(3)입니다. 4는 IV인데, V(5)에서 1(I)를 빼겠다는 뜻이며, 반대로 VI는 5+1=6이 됩니다(V(5)를 기준으로 왼쪽은 뺄셈, 오른쪽은 덧셈이라고 생각하면 쉽습니다). 40과 90은 각각 XL하고 XC인데, 50(L) - 10(X)와 100(C)-10(X)로 표현된 것을 볼 수 있습니다.

로마 숫자는 한국어에서 숫자를 비교하는 것과 부분적으로 비슷합니다.

예를 들면 24를 표현하자면: $24 = 20 + 4 = XX + IV = XXIV$ (10+10-1+5 라고도 볼 수 있습니다) 한국어에서 24를 풀어써 쓰면 "이십 사"가 됩니다. 한국에서 이십과 사를 같이 붙여서 말하듯 로마식에서도 비슷하게 붙여서 씁니다.

$37 = 30 + 7 = XXX + VII = XXXVII$

$692 = 600 + 90 + 2 = DC + XC + II = DCXCII$

더 자세한 내용은 아래 링크를 참고해 주세요

https://en.wikipedia.org/wiki/Roman_numerals

(요구사항)

- (1) 사용자로부터 변환할 10진 정수를 입력 받습니다. 정수의 범위는 1에서 1000까지이며, 이 범위 외의 정수를 입력하면 에러 메시지 출력 후, 다시 입력을 받습니다.
- (2) 범위내의 input이 들어오면 그 숫자가 로마 숫자로 무엇인지 나타내면 됩니다.
- (3) 사용자로부터 로마 숫자로 변환하기를 계속할 것인지 여부를 입력 받아 'y'를 입력 받은 경우에는 변환 할 숫자를 다시 입력 받으며, 'n'을 입력 받은 경우에는 초기 선택 메뉴로 돌아 갑니다. 'y'와 'n' 입력 외에는 고려하지 않습니다.

```
=====
CSED 101 Assignment 1
Choose from the following:

1. Riemann Sum
2. Roman Numerals
3. The Mystery Number
4. Terminate Program
=====
Selection: 2
```

```
Enter number: -123
The number should be positive

Enter number: 100000
The number should be equal to or less than 1000

Enter number: 888
Decimal Number: 888
Roman Numerals: DCCCLXXXVIII

Do you want to continue(y/n)? y

Enter number:
```

(4) 이 문제와 관련하여 아래 사용자 정의 함수를 반드시 구현 후, 사용해야 합니다.

- **void funcRomanNumerals():** 초기 메뉴에서 사용자가 2를 입력한 경우에 실행되는 함수로, 사용자로부터 10진 정수를 입력 받은 후, 아래 printRoman()를 호출하여 해당 기능을 수행하는 함수
- **void printRoman(int num):** 정수 num을 로마 숫자로 변환하여 출력하는 함수

(힌트)

로마 숫자로 변환해줄 함수들, 즉 아래와 같이 각각 1, 10, 100의 자리를 출력하는 함수로 나눠서 해결하는 방법이 있습니다. 이 함수들은 switch 문을 사용하여 해결할 것을 추천합니다.

- **void printOnes():** 1의 자리 숫자를 출력하는 함수
- **void printTens():** 10의 자리 숫자를 출력하는 함수
- **void printHundreds():** 100의 자리 숫자를 출력하는 함수

단, 위 기능 구현 시 switch 문 등을 사용하여 하나하나 1천개의 출력문을 가지도록 작성하는 건 절대 안됩니다.

힌트 외에 다른 방법을 사용해도 됩니다.

IV. 미스터리 숫자 생성 (25점)

(문제) 특정 범위의 정수(범위: 1 ~ 1,000,000,000)를 입력 받아서 미스터리 숫자를 생성합니다.

(설명)

미스터리 숫자는 아래의 순서대로 계산합니다.

- A. 입력된 숫자의 왼쪽에서 오른쪽으로 한 자리씩 추출하여, 제일 왼쪽의 숫자는 1제곱, 그 다음 숫자는 2제곱, ... 마지막 숫자는 n제곱으로 계산하여 모두 합칩니다.

예) 19178 → $1^1 + 9^2 + 1^3 + 7^4 + 8^5 = 35252$

- B. 합친 값이 소수(prime number)이면 원래 숫자에서 가장 큰 digit을 더하고, 소수가 아니면 가장 작은 digit을 뺍니다.

- 원래 숫자에서 가장 큰 digit: 원래 숫자의 각 자리의 숫자를 비교하여 가장 큰 값

예) 35252는 소수가 아니므로 19178에서 가장 작은 숫자에 해당하는 1을 뺍니다: $35252 - 1$

만약 35252가 소수였다면 19178에서 가장 큰 숫자에 해당하는 9를 더합니다: $35252 + 9$

(요구사항)

- (1) 입력되는 정수의 범위는 1 ~ 1,000,000,000까지 입니다. 이 범위 외의 입력 값은 에러 메시지를 출력 후, 다시 입력 받습니다. (정수의 입력 범위를 벗어난 숫자 입력은 고려하지 않습니다.)

입력 정수의 타입은 int 를 사용하면 됩니다. (만약 작은 숫자의 계산은 정확한데 큰 숫자의 계산이 부정확하다고 생각되면 long long int를 사용할 것을 추천합니다.)

- (2) 위에서 설명한 미스터리 숫자 계산하는 방법에 따라 숫자를 계산하여 출력합니다.

- (3) 사용자로부터 '미스터리 숫자 생성'을 계속할 것인지 여부를 입력 받아 'y'를 입력 받은 경우에는 또 다른 숫자를 입력 받아 미스터리 숫자를 생성하고 'n'을 입력 받은 경우에는 초기 메뉴로 돌아갑니다. 'y'와 'n' 입력 외에는 고려하지 않습니다.

- (4) 이 문제와 관련하여 아래의 사용자 정의 함수를 반드시 구현한 후 사용해야 합니다.

- **void funcMysteryNumber():** 초기 선택 메뉴에서 3을 입력한 경우에 실행되는 함수
사용자로부터 정수를 입력 받아 미스터리 숫자를 계산하여 출력하는 기능을 수행하는 함수로 아래 정의된 함수를 실제 호출하여 기능을 수행 함
- 계산된 결과가 소수(prime number)인지 아닌지 알아내는 함수
 - ✓ **int checkPrime(int num):** 정수 num이 소수이면 1을, 소수가 아니면 0을 반환
- 자리 숫자 중 제일 큰 수와 작은 수를 알아내는 함수
 - ✓ **int maxDigit(int num):** 정수 num의 각 자리 숫자 중 가장 큰 숫자를 찾아 반환
 - ✓ **int minDigit(int num):** 정수 num의 각 자리 숫자 중 가장 작은 숫자를 찾아 반환

CSED 101 Assignment 1

Choose from the following:

1. Riemann Sum

```

2. Roman Numerals
3. The Mystery Number
4. Terminate Program
=====
Selection: 3

Enter your number: -12
Positive values only

Enter your number: 1200000000
The number should be equal to or less than 1000000000

Enter your number: 13153
The final result of the input 13153 is: 878

Do you want to continue(y/n)? y

Enter your number: 19178
The final result of the input 19178 is: 35251

Do you want to continue(y/n)? n

=====
CSED 101 Assignment 1
Choose from the following:

1. Riemann Sum
2. Roman Numerals
3. The Mystery Number
4. Terminate Program
=====
Selection:

```