

과제 #2

HW1, HW2, HW3, HW4

과제 제출 마감: 9/29 24:00

조교 노인우, inwoo13@hanyang.ac.kr

조교 한중수, soohan@hanyang.ac.kr

Homework

- ◆ 과제 제출 마감: 9/29 24:00
- ◆ Gitlab repository에 “HW2” 폴더를 만든 후 진행
- ◆ 실습 서버 제출, 개인 PC 제출 중 편한 방법으로 제출
- ◆ 소스코드들을 HW2 폴더 내에 위치시킨 후 제출
- ◆ 채점 기준은 실습서버 환경에서 채점

Homework_01 – “진법변환 (radix notation)”

◆ 입력된 수를 주어진 진법으로 표시하는 프로그램 작성

- 첫 번째 인자는 진법으로 2부터 36까지 가능하다. 진법에 그 외의 값이 입력되면 바로 종료한다.
- 두 번째 인자부터 출력으로 해당 진법으로 표현한다. (0-9, 소문자 a-z를 사용)
- 주어진 skeleton code에서 RadixNotation 함수를 구현한다.
(나머지 부분은 수정하지 않음)

◆ 파일명: radix_notation.cc

◆ 입력: 양의 정수인 k진법과 n개의 숫자 num1, num2 ... numN. (단, $2 \leq k \leq 36$, $n < 20$ 인 자연수, $\text{numN} \leq 65536$ 인 자연수)


◆ 출력: 화면에 변환된 문자열을 개행 하여 출력.

◆ 입출력은 cin, cout 을 이용.

Homework_01 – “진법변환 (radix notation)”

◆ 입력 및 출력 예시

```
$ ./radix_notation 1 123
$ ./radix_notation 10 123
123
$ ./radix_notation 8 10 250
12 .....▶ 10의 8진법 표현
372 .....▶ 250의 8진법 표현
$ ./radix_notation 16 250
fa
$ ./radix_notation 2 250
11111010
$ ./radix_notation 32 4 10 200 255 65536
4
a
68
7v
2000
```



Homework_01 – “진법변환 (radix notation)”

◆ Skeleton code

```
#include <stdio.h>
#include <iostream>
#include <string>
```

```
using namespace std;
```

구현 필요

```
/* Implement this function. */
string RadixNotation(unsigned int number, unsigned int radix);
```

```
int main(int argc, char** argv) {
    if (argc < 2) return -1;
    unsigned int radix;
    sscanf(argv[1], "%u", &radix);
    if (radix < 2 || radix > 36) return -1;
    for (int i = 2; i < argc; ++i) {
        unsigned int number;
        sscanf(argv[i], "%u", &number);
        cout << RadixNotation(number, radix) << endl;
    }
    return 0;
}
```

Homework_02 – “간단한 점 그리기 (draw points)”

- ◆ 입력되는 2차원 좌표를 점으로 화면 출력하는 프로그램 작성
 - 매번 점의 x, y 좌표 (음이 아닌 정수)가 주어진다.
 - 화면에 점이 없는 위치에는 ‘.’을, 점이 있는 위치에는 ‘*’을 출력한다.
 - 좌측 상단은 $(0, 0)$ 으로 시작한다.
(따라서, 오른쪽과 아래쪽 모서리에는 항상 ‘.’이 있어야 한다.)
 - 음수 좌표가 입력되면 프로그램을 종료한다.
- ◆ 파일명: draw_points.cc
- ◆ 입력: 각 점의 x, y 위치
(단, $x \leq 50, y \leq 100$)
- ◆ 출력: 각 점의 위치를 나타내는 도형

Homework_02 – “간단한 점 그리기 (draw points)”

◆ 입력 및 출력 예시

```
$ ./draw_points
3 1 ..... (3, 1) 좌표를 의미
...
* ..... (3, 1) 위치에 * 출력
...
1 2
...
* ..... (3, 1), (1, 2) 위치에 * 출력
* ..
. * ..
0 1
...
* .. *
. * ..
-1 -1 ..... 음수 좌표 입력, 프로그램 종료. 추가 출력 없음.
```

Homework_03 – “소인수분해(Prime Number Factorization)”

- ◆ 자연수를 인자로 입력하면 소인수분해 하여 출력하는 프로그램 작성
 - 가장 작은 소인수부터 차례대로 출력
 - 곱셈 연산자는 \times 로 표현하며, 제곱 연산자는 $^$ 로 표현한다. (e.g. $3^3 \times 5^2$)
 - 1은 소인수가 아니므로 포함하지 않는다.
 - 소수가 입력되었을 시, 소수 1 로 표현한다. (e.g. 19^1)
 - 자연수가 아닌 값(문자열 또는 실수 등)이 입력되면 출력 없이 프로그램을 종료한다.
- ◆ 파일명: prime_factorization.cc
- ◆ 입력: 자연수 n
(단, $n \leq 65535$)
- ◆ 출력: 소인수분해 된 결과를 출력 (연산자간 띄어쓰기에 유의)

Homework_03 – “소인수분해(Prime Number Factorization)”

◆ 입력 및 출력 예시

```
$ ./prime_factorization 5
```

```
5^1
```

```
$ ./prime_factorization 12
```

```
2^2 x 3^1
```

.....▶ 12를 소인수분해 한 결과. x 문자열 전후로 공백 한 칸 출력

```
$ ./prime_factorization 36
```

```
2^2 x 3^2
```

```
$ ./prime_factorization 720
```

```
2^4 x 3^2 x 5^1
```

자연수 12를 인자로 입력

Homework_03 – “소인수분해(Prime Number Factorization)”

◆ Skeleton code

```
#include <iostream>
#include <stdio.h>

using namespace std;

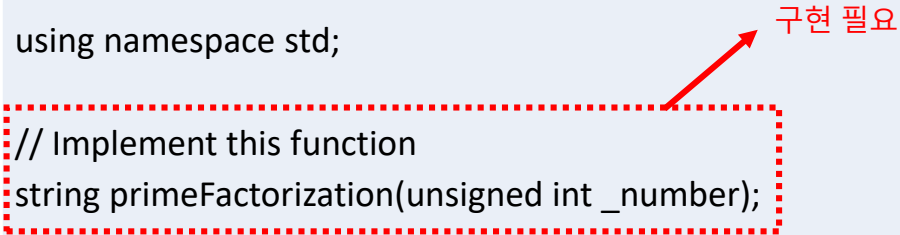
// Implement this function
string primeFactorization(unsigned int _number);

int main(int argc, char** argv)
{
    if(argc < 1)
        return -1;

    unsigned int number;
    sscanf(argv[1], "%u", &number);

    cout << primeFactorization(number) << endl;
    return 0;
}
```

구현 필요



Homework_04- “댓글 관리 프로그램(Reply Administrator)”

◆ 댓글을 입력하고, 인덱스와 명령어를 통해 댓글을 삭제하는 프로그램 작성

- 프로그램을 실행한 뒤, 문자열을 입력하여 댓글을 추가할 수 있다.
(개발의 편의를 위해, skeleton code에 임의로 추가된 댓글이 존재한다)
- 추가된 댓글에는 추가한 순서대로 0 부터 번호를 부여한다.
- 모든 명령어는 '#' 문자로 시작한다.
- #remove 명령을 통해 댓글을 삭제할 수 있다.
 - #remove [Number]: 특정 번호의 댓글을 삭제한다. (e.g. #remove 3)
 - 번호가 존재하지 않을 경우 아무런 내용도 출력하지 않는다.
 - #remove [number]-[number]: 영역 내의 댓글을 모두 삭제한다. (e.g. #remove 2-5)
(단, 영역이 존재하는 댓글을 초과할 경우, 영역 내 존재하는 댓글만 삭제한다)
 - #remove [number],[number], ... ,[number]: 선택된 댓글을 삭제한다. (e.g. #remove 0,1,3)
(단, #remove 0,1,3에서 삭제하는 인덱스는 임의 순서로 주어진다 e.g. #remove 1,0,3)

◆ 파일명: reply_admin.cc

◆ 입력: 단순 문자열 or 명령어가 포함된 문자열 (#remove, #quit) (문자열 길이 <= 126)

Homework_04- “댓글 관리 프로그램(Reply Administrator)”

◆ 입력 및 출력 예시

```
$ ./reply_admin
Hello, World!
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 I think TAs look cool.
6 Hello, World!
#remove 5
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 Hello, World!
```

Hello, World! 입력

미리 주어진 문자열들

추가된 Hello, World! 문자열 출력

5번을 제거하는 명령 입력

기존 목록의 5번이 사라진 것을 확인

```
#remove 0,5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
C++ is so hard
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
Yesterday was too bad.
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
5 Yesterday was too bad.
#remove 4-5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
#quit
```

Homework_04- “댓글 관리 프로그램(Reply Administrator)”

◆ Skeleton code

```
#include <iostream>
#include <string>
#include <stdio.h>

using namespace std;

const int NUM_OF_CHAT = 200;

int getChatCount(string *_chatList)
{
    int i;
    for(i=0; i<NUM_OF_CHAT; ++i)
    {
        string s = _chatList[i];
        if(s.empty() == true) break;
    }
    return i;
}

void printChat(string *_chatList)
{
    int count = getChatCount(_chatList);
    for(int i=0; i<count; ++i)
    {
        cout << i << " " << _chatList[i] << endl;
    }
}
```

구현 필요

```
// Implement these functions
bool addChat(string *_chatList, string _chat); // returns true when adding chat is succeeded
bool removeChat(string *_chatList, int _index); // returns true when removing chat is succeeded
```

// Implement commented (/* */) areas in main function

```
int main(void)
{
    string* chats = new string[NUM_OF_CHAT];

    addChat(chats, "Hello, Reply Administrator!");
    addChat(chats, "I will be a good programmer.");
    addChat(chats, "This class is awesome.");
    addChat(chats, "Professor Lim is wise.");
    addChat(chats, "Two TAs are kind and helpful.");
    addChat(chats, "I think TAs look cool.");
```

미리 추가된 문자열들

```
while(true)
{
    string command;
    getline(cin, command);

    if(/* #quit */) break;
    else if(/* #remove */)
    {
        /* remove chat */
        if(/* remove is succeeded */)
            printChat(chats);
    }
    else if(addChat(chats, command))
        printChat(chats);
}
```

/* */ 들 내부 구현 필요

```
// delete chatting list
delete[] chats;
```

```
return 0;
```

```
}
```

