

창의적 소프트웨어 프로그래밍

실습 과제_03

2-1. 진법 변환(Radix notation)

- 입력된 수를 주어진 진법으로 표시하는 프로그램 작성.
- 첫번째 인자는 진법으로 2에서 36까지 가능하며, 진법에 그 외의 값이 입력되면 그냥 종료한다
- 그 다음 인자들을 해당 진법으로 표현한다 (0-9 a-z 를 사용)
- 주어진 skeleton에서 radixNotation 함수를 구현한다
(나머지 부분은 수정하지 않음)

2-1. 진법 변환(Radix notation)

- 파일명 : radixNotation(radixNotation.cpp)
- **입력** : 양의 정수인 숫자와 진법
- **출력** : 화면에 변환된 문자열을 출력

```
$/radixNotation 16 250
fa
$/radixNotation 2 250
11111010
$/radixNotation 32 4 10 200 255 65536
4
a
68
7v
2000
```

2-1. 진법 변환(Radix notation)

- Code skeleton

```
// radixNotation.cpp
#include <stdio.h>
#include <iostream>
#include <string>
using namespace std;

// 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;
}
```

2-2. 간단한 점 그리기(Draw points)

- 입력되는 2차원 좌표를 점으로 화면에 출력하는 프로그램 작성
 - 매번 점의 x, y 좌표(음이 아닌 정수)가 주어진다
 - 화면에 점이 없는 위치에는 '.'을 점이 있는 위치에는 '*'를 출력한다
 - 좌 상단이 (0,0) 이며 꼭 필요한 부분만 출력한다
즉, 오른쪽과 아래쪽 모서리에는 항상 점이 있어야 한다
 - 좌표로 음수가 입력되면 프로그램을 종료한다

2-2. 간단한 점 그리기(Draw points)

- 파일명 : drawPoints (drawPoints.cpp)
- **입력** : 각 점의 x, y 위치
- **출력** : 각 점의 위치를 나타내는 도형

```
$/drawPoints
3 1
....
...*
1 2
....
...*
.*..
0 1
....
*.*
.*..
-1 -1
$
```

2-3. 소인수분해(Prime number factorization)

- 자연수를 입력하면 자연수를 소인수분해하여 출력하는 프로그램 작성
 - 가장 작은 소인수부터 차례대로 출력
 - 곱셈 연산자는 x로 표현하며, 제곱 연산자는 ^로 표현한다 (e.g. $3^3 \times 5^2$)
 - 1은 소인수가 아니므로 포함하지 않는다
 - 소수가 입력 됐을 시에는, 소수¹로 표현한다(e.g. 19^1)
 - 자연수가 아닌 값(문자열, 실수 등)이 입력되면 출력 없이 프로그램을 종료한다

2-3. 소인수분해(Prime Number Factorization)

- 파일명 : primeFactorization (primeFactorization.cpp)
- **입력** : 자연수
- **출력** : 소인수분해 된 결과를 출력 (연산자 간 띄어쓰기에 유의)

```
./primeFactorization 5
5^1
./primeFactorization 12
2^2 x 3^1
./primeFactorization 36
2^2 x 3^2
./primeFactorization 720
2^4 x 3^2 x 5^1
```


2-3. 소인수분해(Prime number factorization)

- Code skeleton

```
#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;
}
```

제출 양식 및 제출 기한

- 제출 양식

- 학번_파일명.cpp파일을 압축해서 **학번.zip**으로 제출
(ex. **2018120511.zip**)
- 각각 파일명
 - 2018120511_radixNotation.cpp
 - 2018120511_drawPoints.cpp
 - 2018120511_primeFactorization.cpp

- 제출 기한

- 6/27 19:00까지
(추가제출기한 **6/27 19:01~6/28 12:00까지**)

채점 기준

- 총 10점 배점
 - 과제 제출 시 기본점수 1점
 - 각 3점씩(한 문제 없을 때마다 -3점)
 - 추가제출기한 제출 시, 문제당 -1점
- 문제 당 감점 사유(아래 사항 중 가장 큰 감점 점수 적용, 중복 적용x)
 - 빌드 오류 -2
 - 문제 조건 미 충족 -1
 - 결과값 오류 -1
 - 다른 소스 중복 -3