완전 기초 스터디!

다섯 번째 모임

키-파

June 6, 2018

목차

쌓아나가기 (w3-h6)

- 정답자 **2명**
- 다음 식을 이용합니다

$$S = \sum_{i=1}^{n} a_i$$

$$\mu = \frac{S}{n}$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} a_i^2 - \mu^2$$

$$= \frac{n \cdot \sum_{i=1}^{n} a_i^2 - S^2}{n^2}$$

- 중간 계산 과정이 아슬아슬하게 2^{31} 을 넘지 않는 것을 확인해 볼 수 있습니다

쌓아나가기 (w3-h6)

코드 (submitted by Coffeetea):

```
#include <cstdio>
using namespace std;

int main(){
    int i,n,sum=0,sum2=0,average,variance;
    scanf("%d",&n);
    for (i=1;i<=n;++i) {
        int a;
        scanf("%d",&a);
        sum=sum+a;
        sum2=sum2+a*a;
    }
    average=sum/n;
    variance=(n*sum2-sum*sum)/(n*n);
    printf("%d\n%d",average,variance);
    return 0;
}
```

괄호쌍 복구 (w3-h7)

- 정답자 **2명**
- 풀이 : 가능한 경우나 불가능한 경우 모두 맨 처음에 오는 것은 (
  1. 맨 처음 하나를 읽고 무조건 (출력
  2. 열린 괄호의 수를 1로 초기화
  3. 나머지를 읽어서 맨 처음 읽은 것과 같으면 (, 아니면) 출력
  4. 열린 괄호의 수를 1 증가하거나 감소
  5. 중간에 열린 괄호의 수가 음수이거나 끝난 후 열린 괄호의 수가 0이 아니면 `wrong` 출력

괄호쌍 복구 (w3-h7)

코드 (submitted by gratus907, modified):

```
#include <cstdio>
using namespace std;

int main() {
    int n, k = 0;
    int now, cond = 0, left = 1;
    scanf("%d%d", &n, &k);
    printf("(");
    for (int i = 1; i < n; i++) {
        scanf("%d", &now);
        if (now == k) {
            k = now;
            left = left + 1;
            printf("(");
        } else {
            left = left - 1;
            printf(")");
        }
        if (left < 0) {
            cond = 1;
        }
    }
    if ((cond == 0 && left != 0)||cond == 1) {
        printf("\nwrong");
    }
}
```

10진수를 2진수로 (w3-h8)

- 정답자 **2명**
- 풀이:
  1. 처음에 $2^{27} = 134217728$을 변수 b에 저장
  2. n이 b보다 크면 1 출력하고 n에서 b를 뺌, 아니면 0 출력
  3. b를 2로 나누고... 28번 반복
  4. 처음 1이 나오기 전의 0은 출력하지 않아야 함
     - 변수 하나를 더 선언하여 해결할 수도 있고...
     - n $= 0$의 처리를 따로 해 주어야

10진수를 2진수로 (w3-h8)

코드 (submitted by Coffeetea):

```
#include <cstdio>
using namespace std;

int main(){
    int n,i,k,s=0;
    k=1073741824;//=2^30
    scanf("%d",&n);
    for (i=1;i<=36;++i) {
        if (k==0) {
            return 0;
        } else if (n<k) {
            k=k/2;
            if (s==1) {
                printf("0");
            }
        } else if (n>=k) {
            s=1;
            n=n-k;
            k=k/2;
            printf("1");
        }
    }
    return 0;
}
```

제대로 나눗셈 (w3-h9)

- 정답자 **2명**
- 초등학교 때 나눗셈 하던 것을 그대로 구현하면 됩니다
- 예를 들어 1 나누기 6이면:
  1. $10 \div 6 = 1 \cdots 4$
  2. $40 \div 6 = 6 \cdots 4$
  3. $40 \div 6 = 6 \cdots 4$
  4. n번 반복

제대로 나눗셈 (w3-h9)

코드 (submitted by gratus907):

```
#include <cstdio>
#pragma warning(disable:4996), _CRT_SECURE_NO_WARNINGS
using namespace std;

int main()
{
    int a, b, n;
    scanf("%d %d %d", &a, &b, &n);
    printf("%d.", a / b);
    for (int i = 0; i < n; i++)
    {
        a = (a%b) * 10;
        printf("%d", a / b);
    }
}
```

없는 수 찾기 (w3-h10)

- 정답자 **1명**
- 20점 풀이:
  1. 10개의 변수를 선언하고... if문을 잔뜩 이용해서...
  2. 없는 것을 찾아서...
- 50점 풀이:
  1. 다음 수를 이용합니다:

$$S = \sum_{i=1}^{n} 2^{a_i} - \sum_{i=1}^{n-2} 2^{b_j}$$

  2. S를 **높은 자리부터** 보면서 두 수를 구한 다음 뒤집어 출력합니다

없는 수 찾기 (w3-h10)

- 100점 풀이:
  1. 다음 두 수를 이용합니다:

$$S = \sum_{i=1}^{n} a_i - \sum_{j=1}^{n-2} b_j$$

$$V = \sum_{i=1}^{n} a_i^2 - \sum_{j=1}^{n-2} b_j^2$$

  2. $2x^2 - 2Sx + S^2 - V = 0$을 만족하는 x를 모두 출력합니다

배열

- 다음의 코드를 정확히 따라 입력합니다

```
#include <cstdio>
using namespace std;

int main() {
    int a[5], i;
    for (i=0; i<5; ++i) {
        scanf("%d", &a[i]);
    }
    printf("%d\n", a[2]);
    for (i=0; i<5; i += 2) {
        printf("%d ", a[i]);
    }
    return 0;
}
```

- 2 6 9 1 4를 넣어 봅니다

배열 뜯어보기

- `int a[5];`
  - 정수형 **배열 변수** `a`를 만드는데, 크기는 5입니다
- `a[1]`
  - 배열 변수 `a`의 **2번째** 공간입니다
- `i += 2`
  - `i = i + 2`와 같습니다
    - 차이점이 있다면, `i`는 한 번만 계산됩니다
  - 마찬가지로 `-=`, `*=`, `/=`, `%=`, $\cdots$ 등이 있습니다
    - 예외 : `!=`, `>=`, `<=`

배열

- w3-h10의 20점 풀이를 그대로 구현한다고 생각해 보세요 (...)
- 변수 공간을 여러 개 만들 (선언할) 수 있는 방법을 제공합니다
  - (자료형) (변수 이름)`[(크기)]`;
- `int a[5];`

a[0]	a[1]	a[2]	a[3]	a[4]
2	6	9	1	4

- `int a[5];`로 선언했다면,
  - `a[5]`를 쓸 수 없습니다
  - 접근 시 `[]` 안에 있는 식이 평가됩니다
    - `a[i]`, `a[i * 2 + 1]` 같은 구문을 자유로이 쓸 수 있습니다

배열

- 문제는 w5-p1입니다

축하합니다!

- 프로그래밍을 다 배우셨습니다!

축하합니다!

- 프로그래밍을 다 배우셨습니다!
- 어느 언어를 배우나 지금 배운 것들을 표현하는 방법을 연습하는 것
  - sequential, conditional, iterative를 기억하시나요?
- 지금부터는 여러분들의 프로그래밍을 편하게 만드는 것을 배울 것
  - 없어도 간단하게 만들 수 있는 것부터 없으면 지옥을 경험하는 것까지

함수

- 다음의 코드를 정확히 따라 입력합니다

```
#include <cstdio>
using namespace std;

void plus(int a, int b) {
    printf("%d\n", a + b);
}

int main() {
    int a, b;
    scanf("%d%d", &a, &b);
    plus(a, b);
    return 0;
}
```

- 3 5를 넣어 봅니다

함수 뜯어보기

- `void plus(int a, int b)`
  - 두 개의 정수형 **값**을 받고, 아무 것도 돌려주지 않는 함수 `plus`를 만듭니다
    - 정수형은 `int`를 통해, 아무 것도 돌려주지 않음은 `void`를 통해 알 수 있습니다
  - 이후 `{ ··· }`에는 함수의 내용이 옵니다

함수 뜯어보기

- `void plus(int a, int b)`
  - 두 개의 정수형 **값**을 받고, 아무 것도 돌려주지 않는 함수 `plus`를 만듭니다
    - 정수형은 `int`를 통해, 아무 것도 돌려주지 않음은 `void`를 통해 알 수 있습니다
  - 이후 `{ ··· }`에는 함수의 내용이 옵니다
- **`int main()`**
  - 아무 것도 받지 않고, 정수형 **값**을 돌려주는 함수 `main`을 만듭니다
- **`return 0;`**
  - 함수의 반환 값으로 `0`을 돌려 줍니다
  - 함수는 **즉시 종료**됩니다

level of **abstraction**

- 사람은 한 번에 하나만 생각하는 것을 잘함
  - 함수는 하나에 하나씩 생각하기 위한 방법의 하나
- 문제가 너무 복잡하면 한 기능 구현에만 집중하세요

함수

- 문제는 w5-p2입니다

함수

- 문제는 w5-p2입니다
- hint: 함수 안에서 자기 자신을 부를 수도 있습니다

비트 연산자

- 다음의 코드를 정확히 따라 입력합니다

```
#include <cstdio>
using namespace std;

int main() {
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d & %d -> %d\n", a, b, a & b);
    printf("%d | %d -> %d\n", a, b, a | b);
    printf("%d ^ %d -> %d\n", a, b, a ^ b);
    printf("%d << %d -> %d\n", a, b, a << b);
    printf("%d >> %d -> %d\n", a, b, a >> b);
    printf("~%d -> %d\n", a, ~a);
    return 0;
}
```

- 5 3을 넣어 봅니다

비트 연산자

- 여기에 있는 연산자들은 각 비트 별로 행동을 취하는 연산자입니다
- & - bitwise and, 비트가 모두 1이면 1, 아니면 0
- | - bitwise or, 비트가 모두 0이면 0, 아니면 1
- ^ - bitwise xor, 1인 비트의 개수가 홀수이면 1, 아니면 0
- <<, >> - 비트를 왼쪽/오른쪽으로 지정된 횟수만큼 밂
- ~ - bitwise not, 비트가 0이면 1, 아니면 0

비트 연산자

a	0	1	0	1
b	0	0	1	1
a & b	0	0	0	1
a | b	0	1	1	1
a ^b	0	1	1	0

a	0	0	0	0	0	1	0	1
~a	1	1	1	1	1	0	1	0
a << 3	0	0	1	0	1	0	0	0
a >> 3	0	0	0	0	0	0	0	0

- abstraction!
  - a << b == $a \cdot 2^b$
  - a >> b == $\lfloor a/2^b \rfloor$
  - ~a == $-a-1$

비트 연산자

- 문제는 w5-p3입니다