

완전 기초 스테디!

네 번째 모임

키-파

May 11, 2018

- 1 지난 시간 문제 풀이
- 2 반복문
- 3 자료형

숫자 읽기 (w3-h1)

- 정답자 4명
- 한 땀 한 땀... 잘 입력하면 됩니다
- 100점이 안 나오신 분들은 대부분 오타입니다
- 코드 (submitted by **죤정**):

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

int main(){
    int a;
    scanf("%d",&a);
    if (a==0){
        printf("%s","zero");
    }
    ...
    if (a==9){
        printf("%s","nine");
    }
    return 0;
}
```

맥주 99병 (w3-h2)

- 정답자 3명
- 196 이상인 경우는 따로 처리, 나머지는 홀짝에 따라 처리
- 역시 100점이 안 나오신 분들 대부분은 오타입니다
- 코드 (submitted by Coffeetea):

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

int main(){
    int n;
    scanf("%d",&n);
    if (n%2!=0) {
        int x;
        x=100-((n+1)/2);
        ...
    } else {
        int x;
        x=99-(n/2);
        ...
    }
    return 0;
}
```

아름다운 출력 (w3-h3)

- 정답자 2명
- 반복문을 이용해서 출력하라는 대로 출력하면 됩니다
 - 숫자를 칸 수에 맞추어 오른쪽 출력하는 것은 %4d 등의 서식 문자를 이용합니다
- 코드 (submitted by Coffeetea):

```
...
int n,k,i;
scanf("%d\n",&n);
for (k=1;k<=n;++k) {
    printf("+-----");
}
printf("+\n");
for (i=1;i<=n;++i) {
    int a;
    scanf("%d",&a);
    printf("|%5d ",a);
}
printf("|\n");
...
return 0;
}
```

- 정답자 **2명**
- 풀이:
 - ① 정렬되었으면 1, 아니면 0인 변수를 만들고 초기에 1을 대입합니다
 - ② 첫 수는 그냥 받고, 두 번째 수부터 기억해 두었다가 이전 수가 더 크면 정렬됨 변수를 0으로 만듭니다
 - ③ 정렬됨 변수에 따라 Yes / No를 출력합니다

- 코드 (submitted by gratus907):

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

int main()
{
    int n;
    scanf("%d", &n);
    int a, b;
    scanf("%d", &a);
    int check = 0;
    for (int i = 1; i < n; i++)
    {
        scanf("%d", &b);
        if (a > b)
        {
            check = 1;
            break;
        }
        a = b;
    }
    if (check == 0)
    {
        printf("Yes");
    }
    else
    {
        printf("No");
    }
}
```

- 정답자 2명
- 풀이:
 - ① 앞 세 개는 직접 입력받습니다
 - ② 이전 변수 세 개를 저장해 두었다 홀수 - 짝수 - 홀수 - 짝수가 나오면 그때 for문의 변수를 출력합니다
 - ③ 반복문 안에서 이전 변수 세 개를 밀어서 다음 변수를 받을 준비를 합니다

- 코드 (submitted by gratus907):

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

int main() {
    int n;
    scanf("%d", &n);
    int a, b, c, now;
    scanf("%d", &a);
    scanf("%d", &b);
    scanf("%d", &c);
    int condition;
    for (int i = 3; i < n; i++) {
        scanf("%d", &now);
        condition = (a % 2 == 1) && (b % 2 == 0) && (c % 2 == 1) && (now % 2 == 0);
        if (condition == 1) {
            printf("%d ", i + 1);
        }
        a = b;
        b = c;
        c = now;
    }
}
```

쌈아나가기 (w3-h6)

- 정답자 **1명**
- 다음 식을 이용합니다

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

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

$$\begin{aligned}\sigma^2 &= \frac{1}{n} \sum_{i=1}^n a_i^2 - \mu^2 \\ &= \frac{n \cdot \sum_{i=1}^n a_i^2 - S^2}{n^2}\end{aligned}$$

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

- 정답자 **1명**
- 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

10진수를 2진수로 (w3-h8)

- 정답자 **1명**

- 풀이:

- ① 처음에 $2^{27} = 134217728$ 을 변수 b 에 저장
- ② n 이 b 보다 크면 1 출력하고 n 에서 b 를 뺀, 아니면 0 출력
- ③ b 를 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

제대로 나눗셈 (w3-h9)

- 정답자 **1명**
- 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

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:

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

- 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 x 를 모두 출력합니다

- <https://site.thekipa.com/for-loop/>

- 다음의 코드를 정확히 따라 입력합니다

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

int main() {
    double d;
    long long int l;
    scanf("%lf%lld", &d, &l);
    printf("%.3lf\n", d / 5);
    printf("%lld\n", l * 2);
    return 0;
}
```

- 12 100000000000을 넣어 봅니다

- `double d;`
 - 실수형 변수 `d`를 만듭니다
- `long long int l;`
 - 긴 정수형 변수 `l`을 만듭니다
- `scanf("%lf%lld", &d, &l); / printf("%.3lf\n", d / 5); / printf("%lld\n", l * 2);`
 - 입력을 받고 출력을 합니다
 - 실수형은 입출력 시 `"%lf"`, 긴 정수형은 입출력 시 `"%lld"`를 사용합니다

- 컴퓨터는 0과 1밖에 모르기 때문에 자료형이 필요함
 - 아주 큰 정수 4607182418800017408은 아주 작은 실수 1.0과 이진 표현이 같습니다
- 여기에 있는 0과 1을 어떻게 해석할지 알려줌
- 정수형 `int`, 긴 정수형 `long long int`, 실수형 `double` 등이 있음
- 변수 선언은 [자료형] [변수 이름];

- 새로운 자료형이 나왔으면 printf와 scanf도 새 자료형이 왔다는 사실을 알아야 합니다
- "%d"는 (십진수와 동시에) 변수의 자료형이 int임을 알려 주던 것
 - double은 "%lf"
 - long long int는 "%lld"

- 여태까지 우리는 `int`의 연산자를 중심으로 배워 왔습니다
- 연산자를 몽땅 새로 배울 필요는 없고 약간의 차이만 있습니다
 - `double`의 나눗셈은 /만 있고 이것은 소수점까지 나오는 나눗셈입니다
 - `long long int`의 연산자는 2^{63} 까지 더 큰 수를 다룰 수 있습니다

- 문제는 **w4-p1**입니다