

프로그래밍 대회에서 사용하는 C, C++, STL

서울시립대학교 알고리즘 소모임 AL林
컴퓨터과학부 2016920060 최형진

1. C

printf, scanf

문제풀기(20분)

○ 1. C

- 8진수, 10진수, 16진수
- A+B - 4
- 그대로 출력하기
- 그대로 출력하기 2
- 숫자의 합
- 열 개씩 끊어 출력하기

8진수, 10진수, 16진수

소스

```
1  #include <stdio.h>
2  int main()
3  {
4      int n;
5      scanf("%i", &n); 자세한 설명(stackoverflow.com)
6      // %i : 10진수, 8진수, 16진수
7      printf("%d", n);
8  }
```

A+B - 4



```
1  #include <stdio.h>
2  int main()
3  {
4      int a, b;
5      while (scanf("%d %d", &a, &b) == 2)
6          printf("%d\n", a + b);
7      return 0;
8  }
```

그대로 출력하기

소스

```
1  #include <stdio.h>
2  int main()
3  {
4      char s[101]; // \n 제외 모두 입력받음
5      while (scanf("%[^\\n]\\n", s) == 1)
6          printf("%s\\n", s);
7      return 0;
8  }
```

자세한 설명

그대로 출력하기 2

소스

```
1  #include <stdio.h>
2  int main()
3  {
4      char c; // EOF : End Of File
5      while ((c = getchar()) && c != EOF)
6          printf("%c", c);
7      return 0;
8  }
```

숫자의 합

소스

```
1  #include <stdio.h>
2  int main()
3  {
4      int n, sum, temp;
5      sum = 0;
6      scanf("%d", &n);
7      while (n-->0)
8      {
9          scanf("%1d", &temp);
10         sum = sum + temp;
11     }
12     printf("%d", sum);
13     return 0;
14 }
```


열 개씩 끊어 출력하기

[소스](#)

```
1  #include <stdio.h>
2  int main()
3  {
4      char s[101];
5      while (scanf("%10s", s) == 1)
6          printf("%s\n", s);
7      return 0;
8  }
```

2. C++

cin, cout

사칙연산

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int A, B;
6      cin >> A >> B;
7      cout << A + B << endl;
8      cout << A - B << endl;
9      cout << A * B << endl;
10     cout << A / B << endl;
11     cout << A % B << endl;
12     return 0;
13 }
```



N 찍기

```
1  #include <iostream>
2  using namespace std;
3  □ int main()
4  {
5      int N;
6      cin >> N;
7      for (int i = 1; i <= N; i++)
8          cout << i << endl;
9      return 0;
10 }
```



문제풀기(생략)

- 2. C++
- N 찍기
- 생일 출력하기
- 피보나치 수 5
- 사칙연산

3. String

문자, 문자열

단어 길이 재기

```
1  #include <stdio>
2  int main()
3  {
4      char s[100];
5      scanf("%s", s);
6      int i = 0;
7      while (s[i] != '\0')
8          i++;
9      printf("%d", i);
10     return 0;
11 }
```

문자열(String)의 끝에는 널 문자('\0')가 붙는다.

정수의 개수

```
1  #include <stdio.h>
2  int main()
3  {
4      char c;
5      int cnt = 0;
6      while ((c = getchar()) && c != '\n')
7          if (c == ',')
8              cnt++;
9      printf("%d", cnt + 1);
10     return 0;
11 }
```


문제풀기(10분) + 휴식(10분)

○ 4. String

○ 단어의 개수 ←

○ 문자열 분석

○ 정수의 개수 ←

○ 단어 길이 재기 ←

○ 더하기

○ 더하기 2

○ 네 수

STL?

- **S**tandard **T**emplate **L**ibrary
- Container, iterator, Algorithm
- <http://en.cppreference.com/w/>

4. STL Container

pair, vector, deque, map, stack, queue, priority_queue, bitset

pair

- 선언 : `pair<first_type, second_type> p;`
- 저장 : `pair<first_type, second_type> p(data, data);` OR `p=make_pair(data, data);`
- 호출 : `p.first` OR `p.second`
- `#include <utility>`
- 좌표

vector

- 선언 : `vector<data_type> v;`
- 저장 : `v.push_back(data);` OR `v.resize(N); v[i] = data;`
- 호출 : `v[i], v.begin(), v.end();`
- `#include <vector>`
- 동적 배열

Stack vs Queue

stack

- First In, Last Out
- 선언 : `stack<data_type> S;`
- `S.push(X);`
- `S.pop();`
- `S.size();`
- `S.empty();`
- `S.top();`
- `#include <stack>`
- 10828번: 스택

queue

- First In, First Out
- 선언 : `queue<data_type> Q;`
- `Q.push(X);`
- `Q.pop();`
- `Q.size();`
- `Q.empty();`
- `Q.front(); Q.back();`
- `#include <queue>`
- 10845번: 큐

deque

- Double-Ended Queue
- 선언 : `deque<data_type> DQ;`
- `DQ.push_front(X); DQ.push_back(X);`
- `DQ.pop_front(); DQ.pop_back();`
- `DQ.size();`
- `DQ.empty();`
- `DQ.front();`
- `DQ.back();`
- `#include <deque>`
- 10866번: 덱

priority_queue

- 우선순위 큐 : 가장 큰 숫자가 front로 온다.
- 선언 : `priority_queue<data_type> PQ;`
- `PQ.push(X);`
- `PQ.pop();`
- `PQ.size();`
- `PQ.empty();`
- `PQ.front();`
- `PQ.back();`
- `#include <priority_queue>`
- 11279번: 최대 힙, 1927번: 최소 힙, 11286번: 절대값 힙

map

- 인덱스 자료형을 독립적으로 지정해줄 수 있음.
- 자동적으로 정렬된 상태 유지. 정렬할 수 없다면 입력 순서의 반대로 출력된다.
- 선언 : `map<index_type, data_type> MAP;`
- 접근 : 반복자(iterator) 또는 배열 첨자
- `#include <map>`
- 1076번: 저항
- 참고 사이트

bitset

- 비트들 저장하는 배열
- 입출력 때 cin, cout을 써야 함.
- 선언 : `bitset A(SIZE);`
- `#include <bitset>`
- 12813번: 이진수 연산

문제풀기(생략)

○ 3. STL Container

- 덱
- 풍선 터뜨리기
- 에디터
- 중복 빼고 정렬하기
- 숫자 카드
- 숫자 카드 2
- 저항
- 듣보잡
- 스택
- 큐
- 조세퍼스 문제
- 최소 힙
- 이진수 연산

5. STL Algorithm

Count, find, fill, reverse, rotate, swap,
sort, binary_search, lower_bound, upper_bound, min, max, next_permutation

비교적 쉬운 기능들

○ 5. STL Algorithm

- 개수 세기 -> count
- 알파벳 개수 -> count
- 알파벳 찾기 -> find
- 공 넣기 -> fill
- 바구니 뒤집기 -> reverse
- 바구니 순서 바꾸기 -> rotate
- 공 바꾸기 -> swap

sort

- `sort(시작, 끝, 정렬조건);`
- 정렬조건 생략하면 오름차순 정렬.
- `#include <functional>` -> 정렬조건에 `greater<자료형>` 쓰면 내림차순 정렬.
- Vector : `sort(v.begin(), v.end());`
- Array : `sort(a, a + N);`
- 2750번: 수 정렬하기

sort – pair에서

- Pair의 first를 기준으로 정렬한다.
- 11650번: 좌표 정렬하기, 11651번: 좌표 정렬하기 2

sort 정렬조건 지정

- 문제에서 두 개 이상의 정렬조건 제시할 때
- 자료들을 구조체로 받고
- 정렬조건 함수를 작성.
- 1181번: 단어 정렬, 10814번: 나이순 정렬, 10825번: 국영수


```
1 struct member
2 {
3     int age;
4     int order;
5     char name[101];
6 };
```

```
8 bool cmp(const member &u, const member &v)
9 {
10     if (u.age < v.age)
11         return true;
12     else if (u.age == v.age)
13         return u.order < v.order;
14     else
15         return false;
16 }
```

binary_search

- `binary_search`(시작, 끝, 찾고자 하는 값);
- 리턴값은 트루 또는 폴스.
- 숫자 카드

min, max

- $\min(A, B)$; $\max(A, B)$;
- 둘 중 작은 값 리턴. 둘 중 큰 값 리턴.
- 예상 범위의 최소 최대값을 미리 저장해놓고 처리하면 좋다.
- 10818번: 최소, 최대

next_permutation / prev_permutation

- 오름차순 정렬 후 next_permutation -> 가능한 순열들로 차례대로 변함
- 내림차순 정렬 후 prev_permutation -> 가능한 순열들로 차례대로 변함
- 반복문으로 사용
- next_permutation(시작, 끝);
- 10819번: 차이를 최대로

시·林

고맙습니다.