

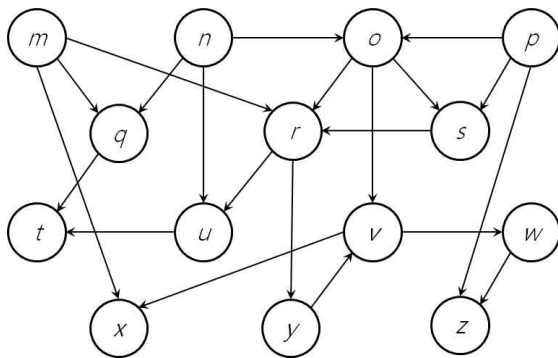
2017 지역대회 (고등부)

[컴퓨팅과 문제해결]

1. (1.1점) 12^{2017} 을 8진수로 표기할 때 가장 오른쪽에 나타나는 연속된 0의 개수는 몇 개일까?

① 673 ② 674 ③ 1344
④ 1345 ⑤ 1556

2. (1.2점) 다음의 그래프에서 모든 간선의 길이가 1일 때 정점 p에서 정점 z까지 도달하는 가장 긴 경로의 길이는?

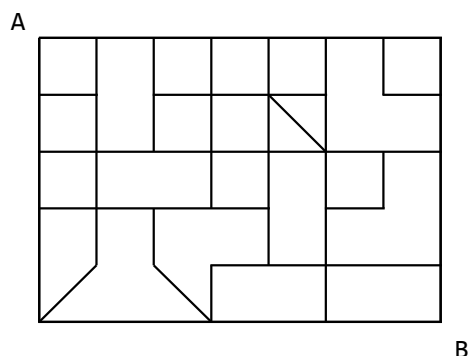


① 7 ② 8 ③ 9
④ 10 ⑤ 11

3. (1.3점) 지구에서 달까지의 거리는 384,400km이다. 여러분에게 폭이 1cm이면서 길이가 충분히 길면서 두께는 0.1mm로 얇은 종이 띠가 주어졌다. 이 종이 띠를 반으로 접으면 접힌 부분의 두께가 두 배로 늘어난다. 최소 몇 번 이상을 접어야 접힌 부분의 두께가 지구에서 달까지의 거리만큼 두꺼워질 수 있을까?

① 38 ② 42 ③ 84
④ 102 ⑤ 420

4. (1.4점) 다음 그림과 같은 도로망에서 A에서 출발하여 B에 도착하는 경로는 몇 가지인가? 단, 오른쪽(→), 아래(↓), 또는 오른쪽 아래 대각선(↘) 방향으로만 이동 가능하다.



① 81 ② 93 ③ 97
④ 129 ⑤ 135

5. (1.5점) A, B, C 세 명이 계단 오르기를 한다. A는 한 걸음에 계단을 1칸 또는 3칸씩 오를 수 있고, B는 한 걸음에 계단을 1칸 또는 4칸씩 오를 수 있고, C는 한 걸음에 계단을 1칸 또는 5칸씩 오를 수 있다고 한다. 세 명은 모두 가능한 최소의 걸음으로 계단을 오른다. 공교롭게도 세 명 모두 같은 횟수의 걸음으로 계단의 가장 위에 도달했다고 한다. 이러한 계단의 최대 칸 수는 얼마인가?

① 15 ② 16 ③ 17
④ 18 ⑤ 19

6. (1.6점) 1번부터 10번까지 번호가 매겨진 10명의 학생이 게임을 하기 위해 팀을 짜려고 한다. 팀은 1명 이상으로 구성되며 한 학생은 반드시 한 팀에만 속해야 한다. 그리고, 어떤 학생 쌍은 사이가 좋아 무조건 같은 팀이 되어야 하고 어떤 학생 쌍은 사이가 좋지 않아 무조건 다른 팀이 되어야 한다. 이에 대한 정보가 아래와 같을 때, 구성될 수 있는 팀의 최소 개수와 최대 개수는 각각 몇 개인가?

- 같은 팀을 원하는 학생 쌍: (1,7), (2,6), (4,5), (6,9)
- 다른 팀을 원하는 학생 쌍: (1,9), (2,8), (4,8), (5,6), (6,10), (7,8), (8,10)

① 3, 5 ② 3, 6 ③ 4, 5
④ 4, 6 ⑤ 팀을 짤 수 있는 방법이 없다.

7. (1.7점) A, B, C, D, E가 처음 만나서 몇 명이 서로 악수를 했다. 이 때, 악수를 한 두 사람의 쌍(순서쌍 아님)들의 집합을 X 라 하자. 적어도 한 쌍이 악수를 했고, 누구도 같은 사람과는 2번 이상 악수를 하지 않았다면 X 가 될 수 있는 집합은 모두 몇 개인가?

① 31 ② 32 ③ 256
④ 1023 ⑤ 1024

8. (1.8점) 철수와 영희는 구슬을 가지고 있다. 철수는 작은 구슬과 중간 구슬을 가지고 있고 영희는 큰 구슬을 가지고 있다. 구슬의 가치는 작은 구슬 9개가 큰 구슬 5개와 같으며, 중간 구슬 9개가 큰 구슬 8개와 같다. 철수와 영희는 서로 같은 가치만큼 구슬을 바꾸려고 한다. 구슬을 자를 수는 없으므로 철수는 작은 구슬 몇 개와 중간 구슬을 몇 개를 합쳐서 영희가 가진 큰 구슬 몇 개와 바꾸어야 한다. 당연히 큰 구슬 1개를 바꿀 수 있는 방법은 없다. 하지만 x 개 이상의 큰 구슬에 대해서는 항상 바꿀 수 있는 방법이 있다.

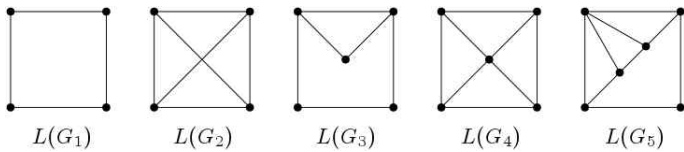
이 때 x 의 최솟값을 구하여라. (철수와 영희는 충분히 많은 수의 구슬을 가지고 있다고 가정하자.)

- ① 2 ② 3 ③ 4
④ 27 ⑤ 28

9. (1.9점) 9명의 남자는 각각 A 개의 구슬을 자신의 주머니에 갖고 있고, 3명의 여자는 각각 B 개의 구슬을 자신의 주머니에 가지고 있다. 남자들이 모두 3명의 여자들에게 각각 C 개의 구슬을 주고, 여자들도 모두 9명의 남자들에게 각각 D 개의 수의 구슬을 주어, 결국 남자와 여자 모두 같은 수의 구슬을 갖도록 하고 싶다. 구슬의 교환은 동시에 이루어진다. 최종적으로 남자와 여자가 가질 수 있는 같은 수의 구슬 개수의 최솟값은 얼마인가? (단, A, B, C, D 는 모두 1 이상의 자연수이고 서로 다르다.)

- ① 9 ② 12 ③ 15
④ 18 ⑤ 21

10. (2.1점) 임의의 단순 무향 그래프 $G=(V,E)$ 의 라인 그래프(line graph) $L(G)=(V',E')$ 는 아래와 같이 정의된다. $V'=E$ 이며, $E'=\{(e,e') \mid e \text{와 } e' \text{는 } G \text{에서 공통된 인접 정점을 갖는다.}\}$ 아래 그림은 어떤 다섯 개의 그래프 G_1, G_2, G_3, G_4, G_5 의 라인그래프를 나타낸 것이다. 이 중에서 원래 그래프가 한붓그림이 불가능한 것은 무엇일까?



- ① G_1 ② G_2 ③ G_3
④ G_4 ⑤ G_5

11. (2.2점) 아래 식에 나타난 각 글자는 2이상 9이하의 서로 다른 한 자리 자연수이다. 아래 식이 성립한다고 할 때, $R+T$ 의 값은 얼마인가?

$$\begin{array}{r} \text{W R O N G} \\ + \text{W R O N G} \\ \hline \text{R I G H T} \end{array}$$

- ① 9 ② 10 ③ 11
④ 12 ⑤ 13

12. (2.3점) 9명의 친구가 각자 차를 몰고 사막의 서쪽 끝에 모였다. 각 차의 연료탱크의 용량은 1리터이고, 리터당 40km을 갈 수 있다. 현재 각 차의 연료탱크에는 1리터가 가득 들어있고, 또한 1리터짜리 연료통 9개가 가득 채워져 각 차에 실려 있다 (단, 트렁크에는 9리터의 연료까지만 실을 수 있고, 한 차에서 다른 차

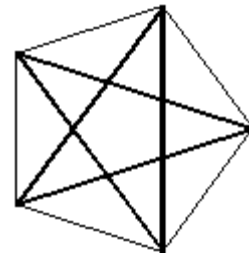
로 연료통을 자유롭게 옮길 수 있다.) 이제 동시에 자신의 차를 운전하여 직선 경로를 따라 동쪽으로 이동했다가 돌아온다. 친구 중 어느 한명이라도 가장 멀리 갔다 돌아오기만 하면 된다. 주의할 것은 반드시 9명의 친구가 자신의 차를 타고 출발지점으로 (동시에 또는 개별적으로) 돌아와야 한다는 것이다. 이 때, 가장 멀리 갈 수 있는 거리는 얼마인가?

- ① 200km ② 240km ③ 280km
④ 320km ⑤ 360km

13. (2.4점) 어떤 함수 $f(n)$ 는 모든 정수 n 에 대하여 $f(n)=f(n-4)+f(n+4)$ 을 만족한다. 만약 $f(1)=1, f(2)=-2, f(3)=-3, f(4)=-1, f(5)=0$ 이라면, $f(2017)$ 은 얼마인가?

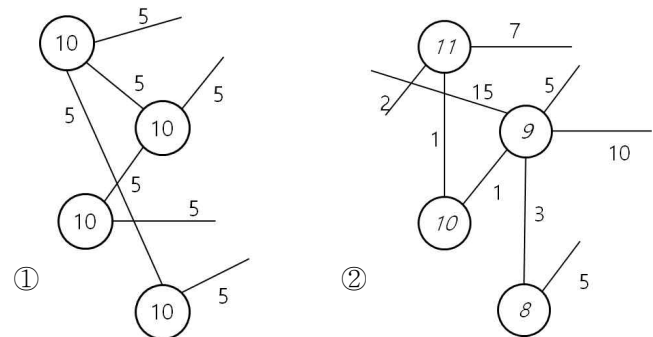
- ① 1 ② -1 ③ 0
④ -2017 ⑤ 2017

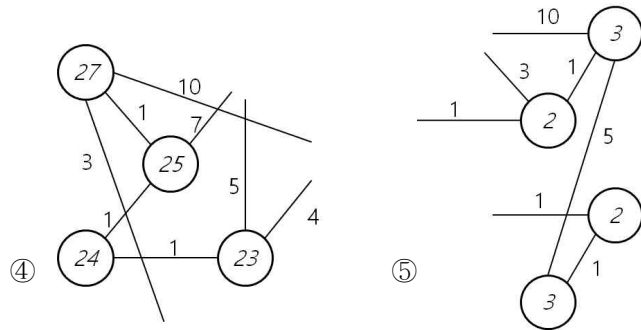
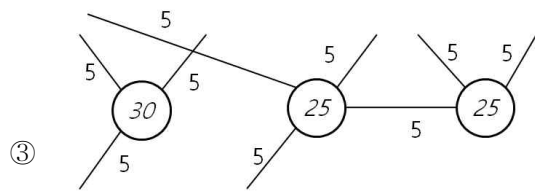
14. (2.5점) 아래 그림에서 발견할 수 있는 삼각형은 모두 몇 개인가?



- ① 25 ② 27 ③ 30
④ 35 ⑤ 37

15. (2.6점) 준우는 서로 다른 5개의 그래프에서 시작 정점에서 모든 정점까지 최단거리를 구하였다. 하지만 급하게 하느라 실수가 있어서 1개의 그래프에 대해서는 최단거리를 잘못 구하였다. 아래와 같이 그래프의 일부만 주어졌을 때 잘못 구한 그래프를 찾아라. 간선 옆에 쓰인 숫자가 간선의 거리이며 각 정점 안에 쓰인 숫자가 준우가 구한 최단거리이다.





[알고리즘과 프로그래밍]

16. (1.0점) 다음 중 C나 C++ 언어의 변수 이름으로 가
능하지 않은 것은?

- ① iff ② brake ③ T_T
④ 0_0 ⑤ _1e9

17. (1.1점) 다음 프로그램의 출력 결과는 무엇인가?
(단, □은 Space을 의미한다.)

```
double a = 2.8437;
printf("%.23lf", a);
```

- ① 2.84 ② 2.844 ③ □□2.84
④ □2.844 ⑤ 2.8437

18. (1.2점) strcpy함수는 src의 문자열을 dest에 복사
하는 함수이다. □(a)에 들어가야 할 내용으로 틀린
것은?

```
void strcpy(char *dest, const char
*src){
    int i;
    for(i = 0; src[i] != □(a); i++)
        dest[i] = src[i];
    dest[i] = □(a);
}
```

- ① '\0' ② 0x0 ③ '0'
④ 0 ⑤ 모두 가능하다

19. (1.3점) 다음 프로그램의 출력은 무엇인가?

```
char a[] = "I^am^sun.";
printf("%d\n", sizeof(a));
```

- ① 1 ② 4 ③ 8
④ 9 ⑤ 10

20. (1.4점) 다음과 같은 함수 f가 구현되어 있을 때
f(5)의 값은 무엇인가?

```
int f(int x)
{
    if (x == 1) return 1;
    return f(x/2) + f((x+1)/2) + 1;
}
```

- ① 5 ② 6 ③ 7
④ 8 ⑤ 9

21. (1.5점) 다음 프로그램의 출력 결과는
무엇인가?

```
int n = 25, s = 0, i;
for (i=1; i<=n; i++) s += n / i;
printf ("%d\n", s);
```

- ① 85 ② 86 ③ 87
④ 88 ⑤ 89

22. (1.5점) 다음 프로그램의 출력 결과는 무엇인가?

```
int i, c = 0;
for(i = 1; i <= 1000; i += 2)
    c += i;
printf("%d", c);
```

- ① 500500 ② 250000 ③ 1000000
④ 1000 ⑤ 1001

23. (1.6점) 다음 프로그램의 출력 결과는 무엇인가?

```
int a, b, c, d, e, result;
a = 3; b = 8; c = 9; d = 10; e = 7;
result = a ^ b + c * c % d + d ^ e;
printf("%d\n", result);
```

- ① 0 ② 1 ③ 23
④ 25 ⑤ 27

24. (1.6점) 다음 프로그램의 출력 결과는 무엇인가?

```
const int l = 7;
const int n = 10;
char S[l+1]="K0I2017";
int p[n] = { 0,5,2,6,7,2,3,1,2,3 };
int c[l+1] = { 0, };
int i,j;
for (i=0;i<n;i++) c[p[i]]++;
for (i=0;i<=l;i++){
    for (j=0;j<c[i];j++)
        putchar('-');
    if (i < l) putchar(S[i]);
}
printf("\n");
```

- ① -K-OI---2--0-1-7-
② K-O-I---2--01-7-
③ -K-O-I--20-1---7-
④ --K-O-I-2-0-1-7--
⑤ -K-O---I--20-1-7-

25. (1.7점) 다음 프로그램의 출력 결과는 무엇인가?

```
void swap(int x, int y){
    int tmp;
    tmp = x; x = y; y = tmp;
}
int main(){
    int a = 3, b = 5, *c = &b, d = 4;
    swap(a, *c);
    c = &d;
    swap(a, *c);
    swap(a, b);
    printf("%d %d %d %d\n", a, b,
    *c, d);
    return 0;
}
```

- ① 3 5 4 4 ② 3 4 5 5 ③ 4 3 5 5
④ 3 5 5 4 ⑤ 5 3 4 4

26. (1.7점) 다음 프로그램의 출력 결과는 무엇인가?

```
const int n = 10;
const int m = 5;
int p[n] = {-1,0,0,1,1,2,2,2,4,4};
```

```
int d[n] = {0,};
int x[m] = {0,6,8,3,8};
int y[m] = {5,7,9,8,5};
int i, res=0;
for (i=1;i<n;i++) d[i] = d[p[i]] + 1;
for (i=0;i<m;i++){
    int a = x[i], b = y[i];
    while (a != b){
        if (d[a] > d[b]) a = p[a];
        else b = p[b];
    }
    res += a;
}
printf ("%d\n",res);
```

- ① 4 ② 5 ③ 6
④ 7 ⑤ 8

27. (1.8점) 다음 프로그램의 출력 결과는 무엇인가?

```
const int n = 10;
int a[n] =
{16,12,17,48,89,21,97,59,30,16};
int i, j, t, res=0;
for (i=0;i<n;i++){
    for (j=i+1;j<n;j++){
        if (a[i] < a[j]){
            t = a[i];
            a[i] = a[j];
            a[j] = t;
        }
    }
}
for (i=0;i<(n/2);i++) res += a[i];
printf ("%d\n",res);
```

- ① 82 ② 182 ③ 223
④ 323 ⑤ 405

28. (1.8점) 다음 프로그램의 출력 결과는 무엇인가?

```
int a[201] = {0};
int i, j;
for (i=1; i<=200; i++) a[i] = i;
for (i=1; i<=200; i++)
    for (j = i + i; j <= 200; j += i)
        a[j] -= a[i];
printf("%d\n", a[180]);
```

- ① 10 ② 48 ③ 39
④ 179 ⑤ 90

29. (1.9점) 다음 프로그램의 출력 결과는 무엇인가?

```
int p[51] = {0,}, i, j, res = 0;
for (i=2;i<=7;i++) if (!p[i]){
    for (j=i*i;j<=50;j+=i) p[j] = 1;
}
for (i=2;i<=50;i++) if (!p[i]){
    int good = 1, t = i;
    while (t){
        if (t % 10 == 4) good = 0;
        t /= 10;
    }
    if (good) res++;
}
printf ("%d\n",res);
```

- ① 12 ② 13 ③ 14
④ 15 ⑤ 16

30. (1.9점) 다음과 같은 함수 f가 구현되어 있을 때 f(123, 4)의 값은 무엇인가?

```
int f(int n, int b) {
    int k = 1, s = 0;
    while (n > 0) {
        s += (n % 10) * k;
        n /= 10;
        k *= b;
    }
    return s;
}
```

- ① 27 ② 57 ③ 108
④ 123 ⑤ 492

31. (2.0점) 다음과 같은 f가 있을 때, f(5,6)을 한 번 호출하면 f(2,2)는 총 몇 번 호출되는가?

```
int f(int x, int y){
    if(x == 0 || y == 0) return 1;
    return f(x - 1, y) + f(x - 1, y - 1) * 2 + f(x, y - 1) * 3;
}
```

- ① 7 ② 129 ③ 139
④ 151 ⑤ 7251

32. (2.0점) 다음 프로그램의 출력 결과는 무엇인가?

```
int res = 0, i;
for (i=0;i<=9999;i++){
    int good = 0, tmp = i;
    while (tmp){
        if (tmp % 100 == 42){
            good = 1;
        }
        tmp /= 10;
    }
    if (good) res++;
}
printf ("%d\n",res);
```

- ① 297 ② 298 ③ 299
④ 300 ⑤ 301

33. (2.1점) 다음 프로그램의 출력 결과는 무엇인가?

```
const int n = 5;
int a[n]={0,1,2,0,2};
int b[n]={1,3,4,3,3};
int k = 2017, i;
char s[n+1] = "01234";
for (i=0;i<k;i++){
    int x = i % n;
    char t = s[a[x]];
    s[a[x]] = s[b[x]];
    s[b[x]] = t;
}
puts(s);
```

- ① 21403 ② 40321 ③ 04231
④ 34102 ⑤ 42301

34. (2.1점) 다음 프로그램의 출력에서 23번째 줄에 있는 숫자는 무엇인가?

```
int i = 7, j;
while (i && i < 128) {
    printf("%d\n", i);
    j = i-(i&(i-1));
    i = (i+j) | (((i+j)^i) / j / 4);
}
```

- ① 29 ② 30 ③ 70
④ 73 ⑤ 127

35. (2.2점) 다음 함수의 리턴값이 5가 되는 양의 정수 n의 개수는 몇 개인가?

```
int f(int n){
    int a[15] = {1, 1, 2, 3, 2, 1, 4,
1, 1, 1, 2, 4, 2, 3, 4};
    int i = 0;
    while(n < 15){
        n += a[n];
        i++;
    }
    return i;
}
```

- ① 1 ② 2 ③ 3
④ 4 ⑤ 5

36. (2.2점) 다음 프로그램의 출력 결과는 무엇인가?

```
int y=2017, i, s = 0;
for (i=y;i; i=(i-1)&y) s += i;
printf ("%d\n",s);
```

- ① 14119 ② 28238 ③ 129088
④ 258176 ⑤ 4068289

37. (2.3점) 다음 프로그램이 1 3 2 3 을 출력하도록 하는 문자열 s의 개수를 구하여라. 단, s는 길이가 10이며, '0' 혹은 '1'로 이루어진 문자열이어야 한다.

```
const int l = 10;
char s[l+1];
int c[4] = {0,}, i;
for (i=1;i<l;i++){
    int a = s[i-1] - '0';
    int b = s[i] - '0';
    c[a * 2 + b]++;
}
for (i=0;i<4;i++) printf ("%d",c[i]);
```

- ① 27 ② 30 ③ 33
④ 36 ⑤ 39

38. (2.3점) 다음 프로그램에서 rook(0)을 호출하면 반환되는 값은 무엇인가?

```
const int n = 6;
int row[n] = {0,}, col[n] = {0,};
```

```
int rook(int a)
{
    if (a == n) return 1;
    int ret = 0, r, c;
    for (r=0;r<n;r++) for
(c=0;c<n;c++){
        if (r == a || c == a)
continue;
        if (row[r] || col[c])
continue;
        row[r] = col[c] = 1;
        ret += rook(a+1);
        row[r] = col[c] = 0;
    }
    return ret;
}
```

- ① 18225 ② 34225 ③ 46225
④ 70225 ⑤ 99225

39. (2.4점) 다음은 정점의 개수가 n개인 그래프에서 플로이드-워셜(Floyd-Warshall) 알고리즘을 구현한 프로그램의 일부이다. d[i][j] 배열은 초기에 정점 i와 정점 j 사이를 잇는 간선의 길이를 담고 있으며, 모든 두 정점 사이에는 간선이 존재한다고 가정한다. 간선의 길이는 양의 정수이다.

아래 프로그램 수행 후, d[i][j] 배열을 i번 정점에서 j번 정점에 이르는 최단 경로의 길이를 담기 위해서 ㉠, ㉡, ㉢에 순서대로 들어갈 내용으로 알맞은 것은 무엇인가?

```
int i, j, k;
for (㉠ = 1; ㉠ <= n; ㉠++) {
    for (㉡ = 1; ㉡ <= n; ㉡++) {
        for (㉢ = 1; ㉢ <= n; ㉢++) {
            if (i != k && j != k && i
!= j) {
                if (d[i][j] > d[i][k]
+ d[k][j])
                    d[i][j] = d[i][k]
+ d[k][j];
            }
        }
    }
}
```

- ① i, j, k
② i, k, j
③ j, k, i
④ j, i, k
⑤ k, j, i

40. (2.4점) 다음 프로그램의 출력값은 무엇인가?

```
char *A[5] = {
    "):=%%=((",
    "\\(><)//",
    "~[,_,]:3",
    ">_>^^<_<",
    "o_o||0_o"
};
int i, j, c = 0;
for(i = 0; i < 5; i++){
    for(j = 0; A[i][j]; j++){
        if( j == 8 ) c++;
    }
}
printf("%d", c);
```

- ① 1 ② 2 ③ 3
④ 4 ⑤ 5

41. (2.5점) 다음은 10000이하의 음이 아닌 정수 $n(n \leq 100)$ 개를 a배열에 입력받아 counting sort를 이용하여 정렬하는 코드이다. 각 빈칸에 들어갈 알맞지 않은 코드는?

```
int i, j, n;
int a[ (1) 100 ], cnt[ (2) 10001 ] = {0, };
scanf("%d", &n);
for(i = 0; i < n; i++){
    scanf("%d", &a[i]);
    cnt[ (3) a[i] ]++;
}
for(i = j = 0; i <= (4) 10000; i++){
    while( (5) cnt[i--] )
        a[j++] = i;
```

- ① (1) : 100 ② (2) : 10001
③ (3) : a[i] ④ (4) : 10000
⑤ (5) : cnt[i--]

[42-43] ※ 다음은 문제와 이 문제를 해결하기 위해 작성한 프로그램의 일부이다.

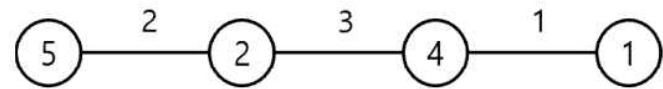
문제

어떤 나라에 N 개의 도시가 있다. 이 도시들은 일직선 도로 위에 있다. 편의상 일직선을 수평 방향으로 두자. 제일 왼쪽의 도시에서 제일 오른쪽의 도시로 자동차를 이용하여 이동하려고 한다. 인접한 두 도시 사이의 도로들은 서로 길이가 다를 수 있다. 도로 길이의 단위는 km를

사용한다.

처음 출발할 때 자동차에는 기름이 없어서 주유소에서 기름을 넣고 출발하여야 한다. 기름통의 크기는 무제한이어서 얼마든지 많은 기름을 넣을 수 있다. 도로를 이용하여 이동할 때 1km마다 1리터의 기름을 사용한다. 각 도시에는 단 하나의 주유소가 있으며, 도시마다 주유소의 리터당 가격은 다를 수 있다. 가격의 단위는 원을 사용한다.

예를 들어, 이 나라에 다음 그림처럼 4개의 도시가 있다고 하자. 원 안에 있는 숫자는 그 도시에 있는 주유소의 리터당 가격이다. 도로 위에 있는 숫자는 도로의 길이를 표시한 것이다.



제일 왼쪽 도시에서 6리터의 기름을 넣고, 더 이상의 주유 없이 제일 오른쪽 도시까지 이동하면 총 비용은 30원이다. 만약 제일 왼쪽 도시에서 2리터의 기름을 넣고 $(2 \times 5 = 10\text{원})$ 다음 번 도시까지 이동한 후 3리터의 기름을 넣고 $(3 \times 2 = 6\text{원})$ 다음 도시에서 1리터의 기름을 넣어 $(1 \times 4 = 4\text{원})$ 제일 오른쪽 도시로 이동하면, 총 비용은 20원이다. 또 다른 방법으로 제일 왼쪽 도시에서 2리터의 기름을 넣고 $(2 \times 5 = 10\text{원})$ 다음 번 도시까지 이동한 후 4리터의 기름을 넣고 $(4 \times 2 = 8\text{원})$ 제일 오른쪽 도시까지 이동하면, 총 비용은 18원이다.

각 도시에 있는 주유소의 기름 가격과, 각 도시를 연결하는 도로의 길이를 입력으로 받아 제일 왼쪽 도시에서 제일 오른쪽 도시로 이동하는 최소의 비용을 계산하는 프로그램을 작성하시오.

입력으로 다음 정보가 주어진다. 첫 번째 줄에는 도시의 개수를 나타내는 정수 $N(2 \leq N \leq 100,000)$ 이 주어진다. 다음 줄에는 인접한 두 도시를 연결하는 도로의 길이가 제일 왼쪽 도로부터 $N-1$ 개의 자연수로 주어진다. 다음 줄에는 주유소의 리터당 가격이 제일 왼쪽 도시부터 순서대로 N 개의 자연수로 주어진다. 제일 왼쪽 도시부터 제일 오른쪽 도시까지의 거리는 1이상 10억 이하의 자연수이다. 리터당 가격은 1 이상 10억 이하의 자연수이다.

프로그램

```
#include <stdio.h>
const int MAXN = 100005;
int n, L[MAXN], P[MAXN];
int main() {
    int i;
    long long p = 1e18, res = 0;
    scanf("%d", &n);
    for (i = 2; i <= n; i++)
        scanf("%d", &L[i]);
    for (i = 1; i <= n; i++) {
        if (i > 1 && p > P[i - 1])
```

```

        (1) ;
    res += (2) ;
    scanf("%d", &P[i]);
}
printf("%lld", res);
}

```

42. (2.5점) 빈 칸 (1)에 들어갈 내용으로 알맞은 것은?

- ① p = P[i]
- ② p += P[i - 1]
- ③ p -= P[i - 1]
- ④ p = P[i - 1]
- ⑤ p *= P[i - 1]

43. (2.6점) 빈 칸 (2)에 들어갈 내용으로 알맞은 것은?

- ① p + L[i]
- ② p + L[i] + 1
- ③ p * L[i]
- ④ p * L[i] + 1
- ⑤ (p + 1) * L[i]

[단답형:컴퓨팅과 문제해결]

44. (2.7점) 이름이 A, B, C, D, E, F, G인 7명이 게임을 한다. 7명은 각자 자신의 주머니에 구슬을 한 개 이상 가지고 있다. 게임에서 이긴 한 사람이 나머지 6명 모두에게 6명 각자 가지고 있는 구슬만큼을 자신이 갖고 있는 구슬에서 주기로 했다. 7번의 게임을 했는데, 이상하게도 모두 한 번씩 정확히 이겼고, 더 이상하게도 이름의 알파벳 순서대로 이겼다. 즉, 첫 게임엔 A가 이겼고, 두 번째 게임엔 B가 이겼고, 이런 식으로 진행되어, 마지막 7번째 게임엔 G가 이겼다. 7번의 게임을 모두 마친 후에 7명 모두 각자가 가지고 있는 구슬의 수가 128개로 동일했다. 그러면 B가 게임을 시작하기 전에 가지고 있던 구슬의 수는 얼마인가? (게임 도중에 구슬이 모자라는 경우는 없었다.)

45. (2.8점) A, B 두 사람이 카드게임을 한다. 이 카드 게임에서는 8장의 카드에 점수가 적혀 있으며, 아래와 같이 일렬로 점수가 보이도록 놓여 있다.

2017 2018 2016 2017 2016 2017 2017 2016
A와 B는 번갈아 가면서 가장 왼쪽 혹은 가장 오른쪽 카드를 가져가며 가져간 카드의 점수만큼을 득점한다. 이를 반복하여 모든 카드가 없어졌을 때, 얻은 점수의

합으로 승패를 가른다. A와 B는 둘 다 모두 전략적으로 플레이하며 따라서 최종 득점을 최대화 하려고 한다. A가 첫 번째로 게임을 시작한다면, A는 몇 점을 득점하게 될까? 1000으로 나눈 나머지를 기입하라.

46. (2.9점) 1번 학생부터 100번 학생까지 100명의 학생이 가위바위보 게임을 한다. 처음에는 모든 학생들이 자기 번호가 쓰인 조끼를 입고 있다. 임의의 두 학생이 승부가 날 때까지 가위바위보를 하여 진 학생은 자기 번호의 조끼를 벗고 이긴 학생과 같은 번호의 새 조끼로 갈아입는다. 또한 가위바위보를 하지 않은 학생 중에서 가위바위보에 진 학생과 같은 번호의 조끼를 입고 있는 학생도 이긴 학생과 같은 번호의 새 조끼로 갈아입는다. (같은 번호의 조끼는 100벌 이상 충분히 많다.) 자기 번호의 조끼를 입고 있지 않은 학생은 더 이상 가위바위보를 할 수 없고 자기 번호의 조끼를 입고 있는 학생들끼리만 가위바위보를 계속할 수 있다. 가위바위보 게임을 반복하여 모든 학생이 같은 번호의 조끼를 입게 된 후 확인해 보니 흥미로운 사실은 가위바위보를 할 때 이긴 학생과 같은 번호의 조끼를 입고 있던 학생의 수가 진 학생과 같은 번호의 조끼를 입고 있던 학생의 수보다 항상 크거나 같았다는 것이다. 이러한 상황에서 한 학생이 입을 수 있는 조끼의 개수의 최댓값은 얼마인가? (처음에 입었던 자기 번호의 조끼도 포함한다.)

[단답형:알고리즘과 프로그래밍]

47. (2.7점) 다음과 같은 함수 f가 주어질 때, f(135) + f(150) + f(165)의 값을 구하여라. 단, gcd(a,b)는 a와 b의 최대공약수를 구하는 함수이다.

```

int gcd(int a, int b)
{
    return b ? gcd(b,a%b) : a;
}

int f(int n)
{
    int res = 0, x;
    for (x=1;x<=n;x++){
        if (gcd(x,n) > 1) continue;
        if (gcd(x+2,n) > 1) continue;
        if (gcd(x+3,n) > 1) continue;
        res++;
    }
    return res;
}

```


48. (2.8점) $r()$ 이 0 혹은 1을 무작위로 반환하는 함수라고 할 때, 다음 프로그램은 12자리의 수를 출력한다. 출력할 수 있는 모든 수 중에서 가장 작은 수의 마지막 세 자리를 구하여라.

```
const int l = 12;
char s[l+1] = "123312332232";
int i = 0, j = l-1;
while (i <= j){
    if (r()) putchar(s[i++]);
    else putchar(s[j--]);
}
```

```
        if (i + 2 * j + 3 * k
== u){
            r += f(i,j,k);
        }
    }
}
return r;
}
```

49. (2.9점) 다음 프로그램이 출력하는 수를 구하여라.

```
const int n = 12;
int b[n] =
{0,3,4,7,8,11,12,15,16,19,20,24};
int p, r = 0;
for (p=0;p<(1<n);p++){
    int x = 0;
    for (int i=0;i<n;i++){
        if (p & (1 << i)) x ^= b[i];
    }
    if (x <= 10) r++;
}
printf ("%d\n", r % 1000);
```

50. (3.0점) 다음 프로그램에서 $g(11)$ 이 반환하는 값을 구하여라. $g(0)$ 은 1을, $g(1)$ 은 1을, $g(2)$ 는 2를 반환하는 것을 참고하라.

```
int f(int x, int y, int z)
{
    if (x < 0 || y < 0 || z < 0)
return 0;
    if (x == 0 && y == 0 && z == 0)
return 1;
    return f(x-1,y,z) + f(x,y-1,z) +
f(x,y,z-1);
}

int g(int u)
{
    int i, j, k, r = 0;
    for (i=0;i<=u;i++){
        for (j=0;j<=u;j++){
            for (k=0;k<=u;k++){
```