Problem 1
Description

A와 B가 마라톤 시합을 한다.
이때, 누가 얼마나 더 빠른지를 구하고,
기록의 차이를 시, 분, 초로 변환하는 프로그램을 작성하시오.

Input
A와 B의 마라톤 기록(r)이 초 단위로 공백으로 구분되어 입력된다.
먼저 입력된 기록이 A의 기록, 나중에 입력된 기록이 B의 기록이다.
1 ≤ r ≤ 86400
Output
빠른 사람, 시, 분, 초를 공백으로 구분하여 출력한다.
시, 분, 초가 0인 경우 0을 출력한다.
기록이 같은 경우 -1을 출력한다.
예를 들어
A가 B보다 51분 30초 빨랐다면
A 0 51 30을 출력한다.

```c
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    if (a < b) {
        printf("A %d %d %d", (b - a) / 3600, ((b - a) % 3600) / 60, (b - a) %
60);
    }

    else if (a > b) {
        printf("B %d %d %d", (a - b) / 3600, ((a - b) % 3600) / 60, (a - b) %
60);
    }

    else
        printf("-1");
}
```

Problem 2
Description

K통신사에는 3가지 요금제가 있다.

A 요금제 : 30초마다 10원이 청구된다. 이 말은 만약 29초 또는 그 보다 적은 시간 통화를 했으면 10원이 청구된다. 만약 30초부터 59초 사이로 통화를 했으면 20원이 청구된다.
B 요금제 : 60초마다 15원씩 청구된다. 이 말은 만약 59초 또는 그 보다 적은 시간 통화를 했으면 15원이 청구된다. 만약 60초부터 119초 사이로 통화를 했으면 30원이 청구된다.
C 요금제 : 120초 마다 25원씩 청구된다. 이 말은 만약 119초 또는 그 보다 적은 시간 통화를 했으면 25원이 청구된다. 만약 120초부터 239초 사이로 통화를 했으면 50원이 청구된다.
통화 목록이 주어졌을 때 가장 저렴한 요금제를 찾는 프로그램을 만드시오.

Input
저번 달에 이용한 통화의 개수 N이 주어진다. 둘째 줄에 통화 시간 N개가 주어진다.
$1 \le N \le 20$
$1 \le$ 통화 시간 ≤ 240
Output
가장 저렴한 요금제와 그때의 가격을 공백을 사이에 두고 출력한다.
만약 가장 저렴한 요금제가 여러개 존재한다면
A, B, C 순서대로 공백을 사이에 두고 출력 후 그때의 가격을 이어서 출력한다.

```c
#include <stdio.h>

int main() {
    int n, x, min = 2147483647;
    int a[3] = {0, };

    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        scanf("%d", &x);
        a[0] += (x / 30) + 1;
        a[1] += (x / 60) + 1;
        a[2] += (x / 120) + 1;
    }

    a[0] *= 10;
    a[1] *= 15;
    a[2] *= 25;

    for (int i = 0; i < 3; i++) {
        if (min > a[i])
            min = a[i];
    }

    for (int i = 0; i < 3; i++) {
        if (min == a[i])
            printf("%c ", 'A' + i);
    }

    printf("%d", min);
}
```

Problem 3
Description

일직선 상에서 앞으로 한칸 또는 뒤로 한칸씩 이동하는 장난감 자동차가 있다.
장난감 자동차는 한번 전원을 켤때마다 앞 또는 뒤로 랜덤하게 8칸을 이동한다.
8칸을 이동하고, 앞 또는 뒤로만 이동한다는 점을 고려하여
전원을 켰을 때 동작을 unsigned char 자료형에 비트 단위로 기록하기로 했다.
1은 앞으로 1칸 이동을 의미한다.
0은 뒤로 1칸 이동을 의미한다.
자동차의 시작 위치는 0이다.

예를 들어
11011101(221) 인 경우
앞 → 앞 → 뒤 → 앞 → 앞 → 앞 → 뒤 → 앞
이동하였으므로 현재 자동차의 위치는 앞으로 4칸이다.

장난감 자동차 4대 A, B, C, D의 전원을 켰을 때,
가장 앞에 있는 장난감 자동차를 구하는 프로그램을 작성하시오.

Input
A, B, C, D 장난감 자동차의 동작이 비트단위로 기록된 값이 공백을 사이에 두고 순서대로 입력된다.
0 ≤ 기록 ≤ 255
Output
가장 앞에 있는 장난감 자동차를 출력한다.
만약 가장 앞에 있는 차가 여러대라면 공백 없이 A, B, C, D 순서대로 출력한다.
예를 들어 가장 앞에 B, C가 있다면
BC출력

```c
#include <stdio.h>

int make_bit(unsigned char n) {
    int count = 0, i = 8;
    while (i--) {
        if (n % 2 == 1)
            count++;
        else
            count--;

        n /= 2;
    }
    return count;
}

int main() {
    unsigned char A, B, C, D;
    int max = -9;
    scanf("%hhu %hhu %hhu %hhu", &A, &B, &C, &D);

    int a[4] = {0, };
    a[0] = make_bit(A);
    a[1] = make_bit(B);
    a[2] = make_bit(C);
    a[3] = make_bit(D);

    for (int i = 0; i < 4; i++) {
        if (max < a[i])
            max = a[i];
    }

    for (int i = 0; i < 4; i++) {
        if (max == a[i])
            printf("%c", 'A' + i);
    }
}
```

Problem 4
Description

애너그램이란 일종의 말장난으로 어떠한 단어의 문자를 재배열하여 다른 뜻을 가지는 다른 단어로 바꾸는 것을 말한다.
https://en.wikipedia.org/wiki/Anagram
예를들어 “adobe”와 “abode”는 서로 길이가 같고, 동일한 문자 구성을 가지고 있기 때문에 애너그램이라고 볼 수 있다.
한 편, dared와 bread는 서로 애너그램 관계에 있지 않다. 하지만 dared에서 맨 앞의 d를 제거하고, bread에서 제일 앞의 b를 제거하면, ared와 read라는 서로 애너그램 관계에 있는 단어가 남게 된다.
두 개의 길이가 같은 영어 단어가 주어졌을 때,
두 단어가 서로 애너그램 관계에 있도록 만들기 위해서 제거해야 하는 최소 개수의 문자 수를 구하는 프로그램을 작성하시오. 문자를 제거할 때에는 아무 위치에 있는 문자든지 제거할 수 있다.

Input
소문자로만 이루어진 단어 2개가 줄바꿈되어 입력된다.
1 ≤ 영단어 ≤ 1000
Output
애너그램 관계에 있도록 만들기 위해서 제거해야 하는 최소 개수의 문자 수를 출력한다.
두 단어에 일치하는 문자가 하나도 없는 경우 -1을 출력한다.
두 단어가 이미 애너그램 관계에 있는 경우 0을 출력한다.

```c
#include <stdio.h>

int main() {
    char a[1000] = {0};
    char b[1000] = {0};
    int a_count[26] = {0};
    int b_count[26] = {0};
    int count = 0;

    scanf("%s", a);
    scanf("%s", b);

    for (int i = 0; a[i] != 0; i++)
        a_count[a[i] - 'a']++;
    for (int i = 0; b[i] != 0; i++)
        b_count[b[i] - 'a']++;

    int none = 1;
    for (int i = 0; i < 26; i++)
        if (a_count[i] != 0 && b_count[i] != 0) {
            none = 0;
            break;
        }

    if (none == 1) {
        printf("-1");
        return 0;
    }

    for (int i = 0; i < 26; i++)
        count += a_count[i] > b_count[i] ? a_count[i] - b_count[i] :
b_count[i] - a_count[i];

    printf("%d", count);
}
```

Problem 5
Description

다음과 같은 비밀번호 규칙이 있을 때, 해당 규칙을 만족하는 비밀번호를 검증하는 프로그램을 작성하시오.

비밀번호는 영문 소문자, 영문 대문자, 숫자, 허용된 특수문자(!@#)로만 구성되어야 한다.
비밀번호의 길이는 9보다 크거나 같고 20보다 작거나 같다.
비밀번호는 최소 2개의 영문 소문자를 포함해야 한다.
비밀번호는 최소 1개의 영문 대문자를 포함해야 한다.
비밀번호는 최소 1개의 숫자를 포함해야 한다.
비밀번호는 최소 1개의 허용된 특수문자(!@#)를 포함해야 한다.
비밀번호는 3개의 연속된 값을 포함하면 안된다.

Input
첫째줄에 비밀번호가 입력된다.
비밀번호는 공백을 포함하지 않는다.
1 ≤ 입력되는 비밀번호 길이 ≤ 30

Output
비밀번호 검증이 통과되었다면 0을 출력한다.
비밀번호 검증이 통과되지 않았다면 실패한 규칙을 순서대로 공백 없이 모두 출력한다.

예를 들어
abc111@? 이 입력된 경우,
허용되지 않은 특수문자 ?가 포함되었다.(1번 규칙 위반)
길이가 9보다 작다(2번 규칙 위반)
영문 대문자를 포함하지 않는다(4번 규칙 위반)
연속된 3개의 값인 111이 포함되었다(7번 규칙 위반)
따라서 출력은 1247이다.

```c
#include <stdio.h>

int main() {
    char pw[100] = {0};
    int check[8] = {0, 1, 0, 0, 0, 0, 0, 1};
    scanf("%s", pw);

    int length = 0;
    while (pw[length])
        length++;

    if (9 <= length && length <= 20)
        check[2]++;

    for (int i = 0; i < length; i++) {
        if ('a' <= pw[i] && pw[i] <= 'z')
            check[3]++;
        else if ('A' <= pw[i] && pw[i] <= 'Z')
            check[4]++;
        else if ('0' <= pw[i] && pw[i] <= '9')
            check[5]++;
        else if (pw[i] == '!' || pw[i] == '@' || pw[i] == '#')
            check[6]++;
        else
            check[1]--;

        if (i >= 2 && pw[i] == pw[i-1] && pw[i-1] == pw[i-2])
            check[7]--;
    }

    int correct = 1;

    for (int i = 1; i <= 7; i++) {
        if (check[i] == 0) {
            if (i == 3 && check[i] < 2)
                printf("3");
            else
                printf("%d", i);

            correct = 0;
        }
    }

    if (correct == 1)
        printf("0");
}
```

Problem 6

Description

입력된 높이를 갖는
모래시계 모양을 ‘*’ 문자를 이용하여 출력한다.
모래시계의 모래는 아래 쪽에만 차있다.

Input

높이(H)는 홀수만 입력된다.
5 ≤ H ≤ 99

Output

별(*) 문자를 이용하여 H 만큼의 모래가 아래 쪽에만 있는 모래시계를 만든다.
자세한 출력은 Sample Output을 참고한다.
출력 후에는 바로 줄바꿈을 해야 함
height가 5인 경우 공백 대신 #을 넣었다고 가정하면

*****
#*#*
##*
#***
*****

위와 같은 형태로 나와야 함
실제 정답 제출 시에는 #이 아닌 공백으로 제출할 것

```c
#include <stdio.h>

int main() {
    int h;
    scanf("%d", &h);

    for (int i = 1; i <= h; i++) {
        if (i == 1 || i == h) {
            for (int j = 1; j <= h; j++)
                printf("*");
            printf("\n");
        }

        else if (i <= h / 2) {
            for (int j = 1; j < i; j++)
                printf(" ");
            printf("*");
            for (int j = i+1; j <= h-i; j++)
                printf(" ");
            printf("*\n");
        }

        else if (i > h / 2) {
            for (int j = 1; j <= h-i; j++)
                printf(" ");
            for (int j = h-i+1; j <= i; j++)
                printf("*");
            printf("\n");
        }
    }
}
```

Problem 7

Description

K고등학교에는 A반부터 J반까지 10개의 반이 있다.
점심시간이 되어 K고등학교 학생들이 운동장에 일렬로 나와있다.

일렬로 나와있는 학생들을 보면
BBCCCAAAADD 인 경우
A, B, C, D 반 학생들이 각각 하나의 그룹으로 모여있기 때문에
학생들이 반 별로 모여있다고 볼 수 있다.

AAABBAADDBB 인 경우
A반 학생과 B반 학생이 두그룹으로 모여있기 때문에
학생들이 반 별로 모여있다고 볼 수 없다.

학생들이 모여있는 입력이 주어질때,
학생들이 반별로 모여있는지를 판단하는 프로그램을 작성하시오.

Input

첫째 줄에 운동장에 일렬로 모여있는 학생들이 입력된다.
운동장에 모여있는 학생의 수는 1보다 크고 100보다 작다.
운동장에 한명도 나오지 않은 반이 있을 수 있다.

Output

학생들이 반별로 모여있는 경우 1을 출력한다.
학생들이 반별로 모여있지 않는 경우 0을 출력한다

```c
#include <stdio.h>

int main() {
    char s[100] = {0};
    scanf("%s", s);

    int check[10] = {0};
    for (int i = 0; s[i] != 0; i++) {
        if (i == 0 || s[i-1] != s[i])
            check[s[i] - 'A']++;
    }
    for (int i = 0; i < 10; i++)
        if (check[i] >= 2) {
            printf("0");
            return 0;
        }
    printf("1");
}
```

Problem 8

Description

특정 날짜로 부터 N-1일이 지난 날짜를 N일 기념일이라고 한다.

예를 들어

2022년 4월 29일의 1일 기념일은 2022년 4월 29일 당일이며,

2022년 4월 29일의 1000일 기념일은 2025년 1월 25일이다.

네이버의 기념일 계산기의 동작과 동일하다.

(https://search.naver.com/search.naver?where=nexearch&sm=top_sug.pre&fbm=0&acr=1&acq=기념일&qdt=0&ie=utf8&query=기념일+계산기)

날짜가 주어질 때, 1000일 기념일을 계산하는 프로그램을 작성하시오.

윤년 규칙은 그레고리력의 윤년 규칙을 따른다.

https://ko.wikipedia.org/wiki/윤년

존재하지 않는 날짜는 입력되지 않는다.

따라서 날짜 검증 로직은 추가할 필요가 없다.

예) 2022년 4월 31일

Input

첫째줄에 year, month, day 가 공백으로 구분되어 입력된다.

1970 ≤ year ≤ 2021

1 ≤ month ≤ 12

1 ≤ day ≤ 31

Output

1000일 기념일을 연, 월, 일 순으로 공백으로 구분하여 출력한다.

```c
#include <stdio.h>

int is_leap_year(int n) {
    return (n % 4 == 0 && n % 100 != 0) || (n % 400 == 0) ? 1 : 0;
}

int main() {
    int general_date[13] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
31};
    int leap_date[13] = {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int year, month, day;
    int count = 1;

    scanf("%d %d %d", &year, &month, &day);

    while (count < 1000) {
        count += is_leap_year(year) ? leap_date[month++] :
general_date[month++];
        if (month == 13) {
            month = 1;
            year++;
        }
    }

    while (count > 1000) {
        count--;
        day--;
        if (day == 0) {
            day = is_leap_year(year) ? leap_date[--month] : general_date[--
month];
            if (month == 0) {
                month = 12;
                year--;
            }
        }
    }

    printf("%d %d %d", year, month, day);
}
```