
02 条件分岐

目標

- 様々な条件式が書けるようになる.
- `if` 文を使えるようになる.
- 適切な場合分けを考えられるようになる.

例題

【電子レンジ】

出典：JOI 2016/2017 予選 A

問題

JOI 君は食事の準備のため、 A °C の肉を電子レンジで B °C まで温めようとしている。肉は温度が 0 °C 未満のとき凍っている。また、温度が 0 °C より高いとき凍っていない。温度がちょうど 0 °C のときの肉の状態は、凍っている場合と、凍っていない場合の両方あり得る。

JOI 君は、肉の加熱にかかる時間は以下のようにになると仮定して、肉を温めるのにかかる時間を見積もることにした。

- 肉が凍っていて、その温度が 0 °C より小さいとき： C 秒で 1 °C 温まる。
- 肉が凍っていて、その温度がちょうど 0 °C のとき： D 秒で肉が解凍され、凍っていない状態になる。
- 肉が凍っていないとき： E 秒で 1 °C 温まる。

この見積もりにおいて、肉を B °C にするのに何秒かかるかを求めよ。

入力

入力は 5 行からなり、1 行に 1 個ずつ整数が書かれている。

- 1 行目には、もともとの肉の温度 A が書かれている。
- 2 行目には、目的の温度 B が書かれている。
- 3 行目には、凍った肉を 1 °C 温めるのにかかる時間 C が書かれている。
- 4 行目には、凍った肉を解凍するのにかかる時間 D が書かれている。
- 5 行目には、凍っていない肉を 1 °C 温めるのにかかる時間 E が書かれている。

もともとの温度 A は -100 以上 100 以下、目的の温度 B は 1 以上 100 以下であり、 $A \neq 0$ および $A < B$ を満たす。

温めるのにかかる時間 C, D, E はすべて 1 以上 100 以下である。

出力

肉を B °C にするのにかかる秒数を 1 行で出力せよ。

入力例 1

-10
20
5
10
3

出力例 1

120

入出力例 1 では、もともとの肉は -10°C で凍っている。かかる時間は以下ようになる。

- -10°C から 0°C まで温めるのに $5 \times 10 = 50$ 秒。
- 0°C の肉を解凍するのに 10 秒。
- 0°C から 20°C まで温めるのに $3 \times 20 = 60$ 秒。

したがって、かかる時間の合計は 120 秒である。

考察

- 最初肉が凍っている場合、0 度まで温めて、解凍して、 B 度まで温めるだけの時間がかかる。
- 最初肉が凍っていない場合、 B 度まで温めるだけの時間がかかる。
- 肉がもともと凍っているか凍っていないかによって答えの計算方法が異なるから、場合分けが必要である。
- いきなりソースコードを記述せずに、まずは文字と日本語を用いて場合分けをする。次の考察の空欄に適切な式などを書き入れよう。

(i) 肉が凍っている、つまり _____ のとき：

肉を 0 度にするのに _____ 秒。

肉を解凍するのに _____ 秒。

肉を目的の温度にするのに _____ 秒。

よって、(i) のときの解は _____ 秒。

(ii) 入力の条件より肉のもとの温度 A は _____ 度ではない。

(iii) 肉が凍っていない、つまり _____ のとき：

肉を目的の温度にするのに _____ 秒。

よって、(iii) のときの解は _____ 秒。

解答例

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int A, B, C, D, E;
    cin >> A >> B >> C >> D >> E;
    int answer;
    if(A < 0){ // kootteiru
        answer = -A * C + D + B * E;
    }
    else if(A > 0){ // kootteinai
        answer = (B - A) * E;
    }
    cout << answer << endl;
}
```

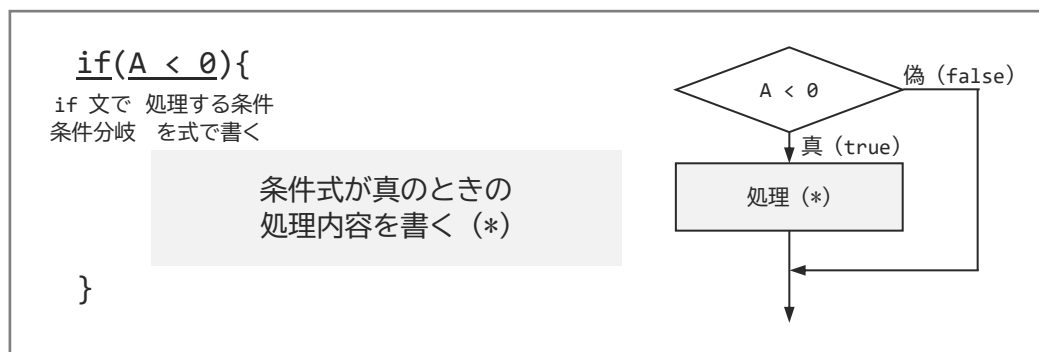
解説

① if 文

条件に基づいた分岐処理を実装するには **if 文** を用いる。if(条件式){ 処理 } の形で記述することで、条件式が満たされる場合のみ { } 内の処理が実行される。

```
if(A < 0){
    answer = -A * C + D + B * E;
}
```

この場合、条件式「 $A < 0$ 」が真 (true) であるときのみ、処理「 $\text{answer} = -A * C + D + B * E;$ 」が実行される。



② 比較演算子

2 つの値を比較する形の条件式を書いてみよう。条件式を書くのに使用可能な **比較演算子** は次の通りである。

比較演算子	意味	比較演算子	意味
$A == B$	A と B が等しい	$A != B$	A と B が異なる
$A < B$	A が B より小さい	$A > B$	A が B より大きい
$A <= B$	A が B 以下	$A >= B$	A が B 以上

代入と区別するため、比較演算子の等号は 2 つ重ねることに注意する。以上、以下ともに等号は後につけることに注意する。

③ 論理演算子

比較演算子は 2 つの値を比較する専用で、3 つ以上の値を 1 つの条件式で比較することはできない。複数の条件式を用いて分岐を行いたい場合は次の **論理演算子** を使う。

論理演算子	意味
$\alpha \ \&\& \ \beta$	α かつ β
$\alpha \ \ \beta$	α または β

例えば A が 0 以上 10 未満であるかを判定したい時、 $0 \leq A < 10$ とは書けないので、A が 0 以上かつ A が 10 未満と分けて書くと、次のようになる。

```
if(0 <= A && A < 10){  
  
}
```

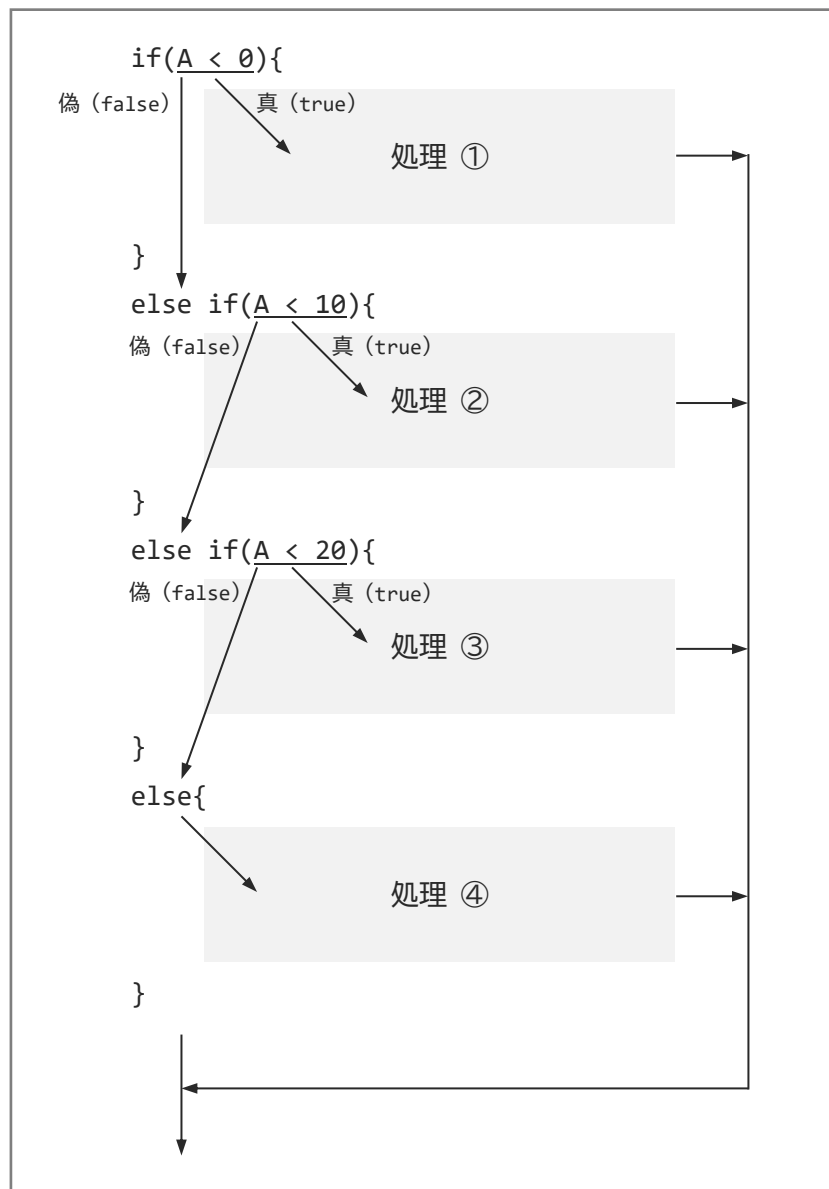
④ else if 文と else 文

複数の条件分岐を同時に行うには **else if 文** や **else 文** を用いる。上から順に条件式が評価され、条件式を満たした場合にはその処理を行って、分岐を抜ける。

```
if(A < 0){  
    // A < 0  
}  
else if(A < 10){  
    // 0 <= A && A < 10  
}  
else if(A < 20){  
    // 10 <= A && A < 20  
}  
else{  
    // 20 <= A  
}
```

例えば上のように記述した場合、

- 「A < 0」が真であるとき、if 文内の処理のみが実行される。
- 「A < 0」が偽でありかつ「A < 10」が真であるとき、1 つめの else if 文内の処理のみが実行される。
- 「A < 0」と「A < 10」がともに偽でありかつ「A < 20」が真であるとき、2 つめの else if 文内の処理のみが実行される。
- 「A < 0」、「A < 10」、「A < 20」がすべて偽であるとき、else 文内の処理のみが実行される。



⑤ コメント

スラッシュ 2 つを連続して `//` と書くと、その行の後の部分がすべて **コメント化** される。コメントはプログラムの実行時には無視される。

```
if(A < 0){ // kootteiru
```

コメント機能を用いて、今回の解答例のように分岐条件を日本語で記述したり、変数の意味をメモしたりすることでソースコードを読みやすくできる。

なお、`/*` と `*/` で囲んだ部分もコメント化され、この場合は複数の行にまたがることもできる。（`*` はアスタリスク）

```

if(A < 0){
    /* kootteiru toki
    kotae ha -A * C + D + B * E */
    answer = -A * C + D + B * E;
}

```

表現

(1) 標準入力から整数 A を受け取り, 絶対値 $|A|$ を出力する

```
int A;
cin >> A;
if(A >= 0){
    cout << A << endl;
}
else{
    cout << -A << endl;
}
```

(2) 標準入力から整数 A, B, C を順に受け取り, それらがすべて等しいなら 1 を, そうでないなら 0 を出力する

```
int A, B, C;
cin >> A >> B >> C;
if(A == B && B == C){
    cout << 1 << endl;
}
else{
    cout << 0 << endl;
}
```

(3) 標準入力から整数 A, B, C を順に受け取り, その最小値を出力する

```
int A, B, C;
cin >> A >> B >> C;
if(A <= B && A <= C){
    cout << A << endl;
}
else if(B <= C){
    cout << B << endl;
}
else{
    cout << C << endl;
}
```

類題

- ☐ Rounding (AtCoder Beginner Contest 130 A)
- ☐ 帰省 (JOI 2020/2021 一次予選 (第 1 回) A)
- ☐ 3 つの整数 (JOI 2019/2020 一次予選 (第 1 回) A)
- ☐ X に最も近い値 (JOI 2019/2020 一次予選 (第 3 回) A)
- ☐ 鉛筆 (JOI 2017/2018 予選 A)