

Training 4

判断と制御

株式会社イーシーエス 出版事業推進委員会

Lesson1 フローチャート



Point◆◇プログラムの処理手順を意識しよう!!

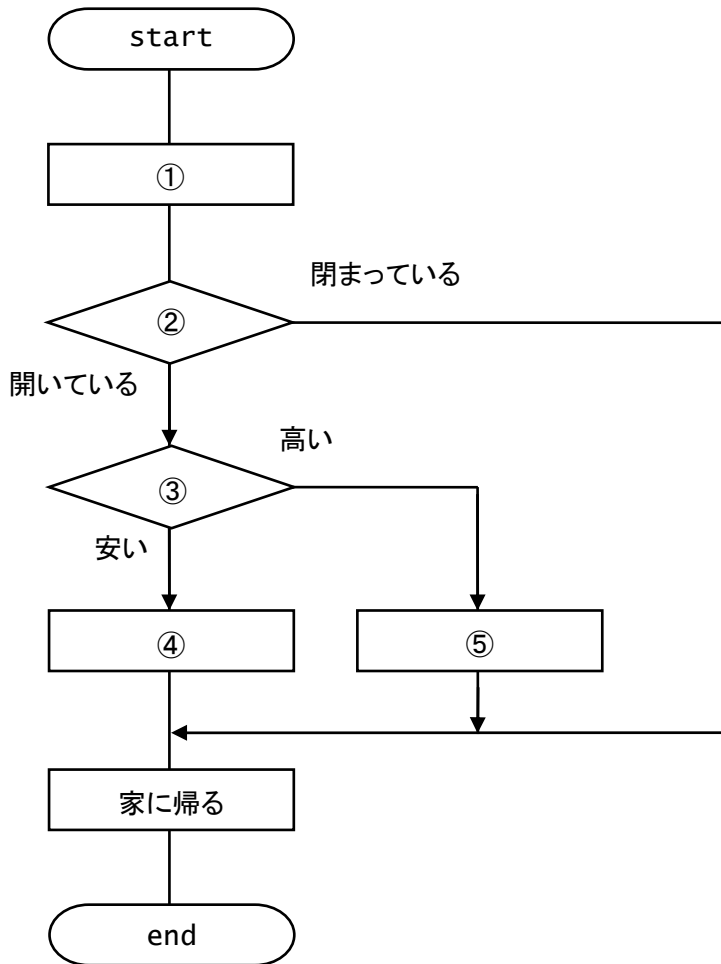
フローチャートはプログラミングなどの処理の手順を各仕事単位に分割して、流れを図で表したものです。

※フローチャートの書き方については巻末資料参照

【問題1】 ①～⑤にあてはまる文章を解答群より選びフローチャートを完成させなさい。

フローチャートは、以下の行動を表したものです。

- ・パン屋に行く
- ・食パンとフランスパンのうち、どちらか安い方を買うこととする。
- ・ただし、お店が休みの場合は家に帰る。

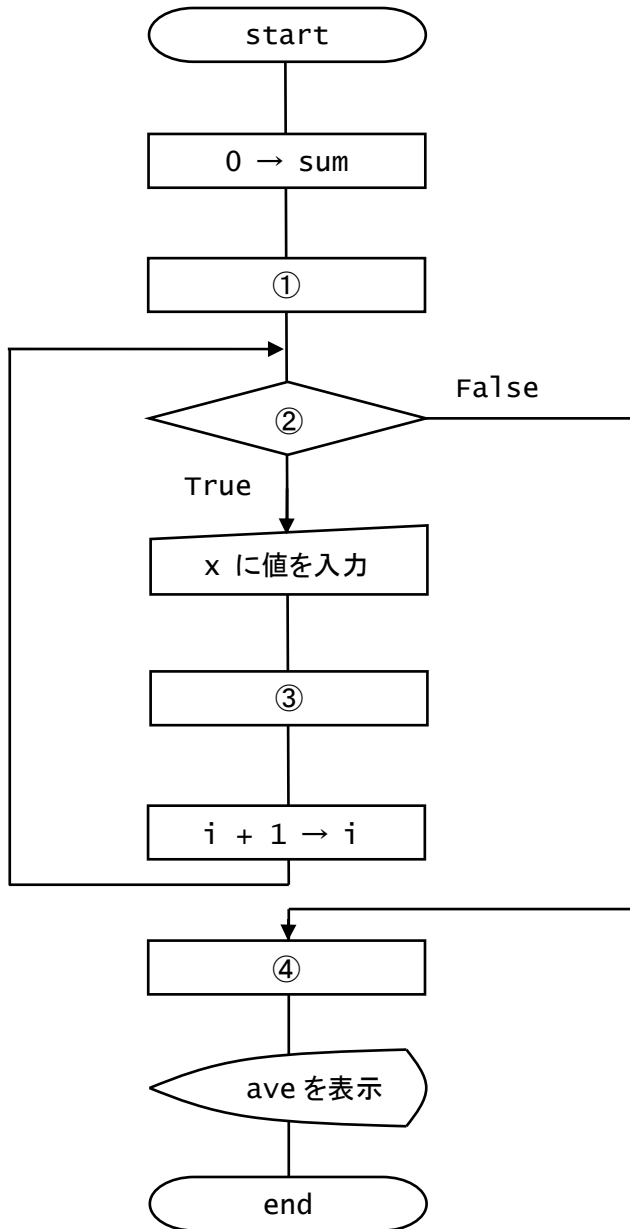


解答群

- ア. 食パンを買う
- イ. フランスパンを買う
- ウ. メロンパンを買う
- エ. 食パンの方が安いかな？
- オ. パン屋が開いているかな？
- カ. パン屋に行く

【問題2】 ①～④にあてはまる式を解答群から選びフローチャートを完成させなさい。

フローチャートは、以下の行動を表したものです。
・6つの値を入力して、その平均を表示する。

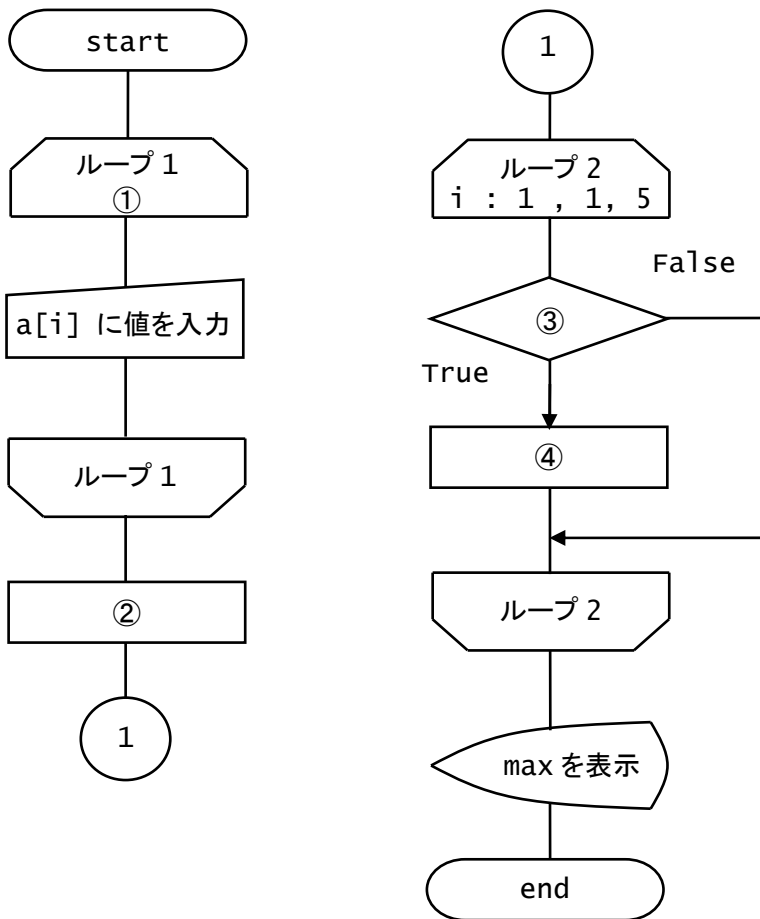


解答群

- ア. $sum + x \rightarrow sum$
- イ. $0 \rightarrow i$
- ウ. $i \leq 5$
- エ. $sum / 6 \rightarrow ave$
- オ. $i \leq 6$

【問題3】 ①～④にあてはまる式を解答群から選びフローチャートを完成させなさい。

フローチャートは、以下の行動を表したものです。
 ・6つの値を配列に入力して、そのなかで一番大きい値を表示する。



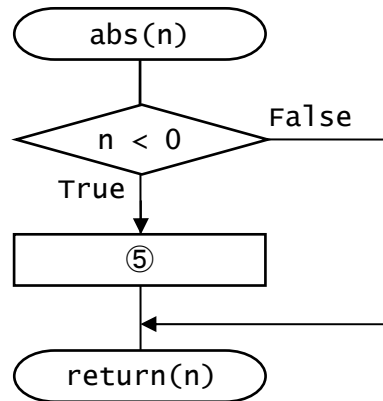
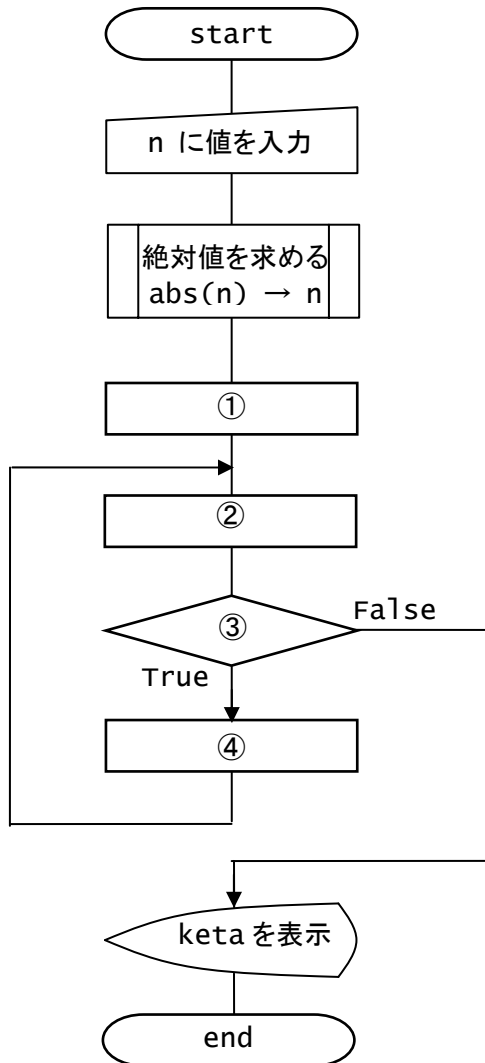
解答群

- | |
|----------------------------|
| ア. $a[0] \rightarrow \max$ |
| イ. $a[i] \rightarrow \max$ |
| ウ. $\max < a[i]$ |
| エ. $i : 0, 1, 5$ |
| オ. $i : 0, 1, 6$ |

【問題4】 ①～⑤にあてはまる式を解答群から選びフローチャートを完成させなさい。

フローチャートは、以下の行動を表したものです。

- ・入力した整数の桁数を求めて表示する。
- ・ただし、負の数の際「-」を桁数に含めないこととする



解答群

- ア. $n * -1 \rightarrow n$
- イ. $1 \rightarrow keta$
- ウ. $n > 0$
- エ. $n = 0$
- オ. $keta + 1 \rightarrow keta$
- カ. $n / 10 \rightarrow n$

【問題5】 以下の処理内容を表すフローチャートを完成させなさい。

【処理内容】

1 から 100 までの和を求めて表示する。

ただし、以下の要素をもとにフローチャートを作成すること。

- ・フローチャートは start から始まり end で終わる。
- ・和の合計値をあらわす変数名を sum とする。
- ・1 から 100 までをあらわす変数名は n とする。
- ・sum を初期化する(初期値 0)。
- ・n を初期化する(初期値 1)。
- ・n が 100 以下かどうか判定する。
- ・sum と n の和を sum に代入する。
- ・n をインクリメントする。
- ・合計を表示する。

【問題6】 以下の処理内容を表すフローチャートを完成させなさい。

【処理内容】

入園料金が1人 500 円の動物園がある。

10 人以上の場合は、1 人あたりの料金が1割引となり、30 人以上の場合は1 人あたりの料金が2 割引になる。

入園する人数を入力値とし、入園料金の合計を表示する。

Lesson2 比較演算



Point◆◇比較演算を理解しよう!!

比較演算は2つの数値を比較して、その比較結果が正しければ真(0以外)、間違っていれば偽(0)を返します。
プログラムでは二つの数値を比較して分岐させたい場合に使われます。

【問題1】 次の□部分に比較演算子を記述して、条件式を完成させなさい。

① Xが4以上のとき

条件式 X □ 4

② Xが5以下のとき

条件式 X □ 5

③ Xが3未満のとき

条件式 X □ 3

④ Xが9より大きいとき

条件式 X □ 9

⑤ Xが8と等しいとき

条件式 X □ 8

⑥ Xが10以外のとき

条件式 X □ 10

【問題2】 以下の条件式の結果が真か偽か答えなさい。

変数情報 X = 5 , Y = 10 , Z = 1

① (X > 5)

② (Y < 11)

③ (Z >= 1)

④ (Y / X > 3)

⑤ (Z == Y < X)

⑥ (((Y - X) * 2) == Y)

⑦ (6 == X + Z)

⑧ (X > Y != X < Z)

Lesson3 論理関係演算



Point◆◇論理関係演算を理解しよう!!

論理関係演算は複数の条件を指定して判定するときに使用します。
使用するときは、各条件式を括弧でくるとプログラムが見やすくなり、バグの減少にもつながります。

【問題1】 次の□部分に論理演算子を記述して、条件式を完成させなさい。

- ① Xが5以上かつ10未満

条件式 ((X >= 5) □ (X < 10))

- ② Xが3より大きいか、またはYが6以上

条件式 ((X > 3) □ (Y >= 6))

- ③ Xが6と同等ではない

条件式 (□ (X == 6))

- ④ Xが20より大きいかつ60以下、またはYが40以上

条件式 (((X > 20) □ (X <= 60)) □ (Y >= 40))

【問題2】 以下の論理演算の結果が真か偽か答えなさい。

変数情報 X = 3 , Y = -2 , Z = 8

- ① ((X > 6) && (Y < 0))

- ② ((Y > 6) || (Z = 7))

- ③ !(Z > 6)

- ④ ((Z <= 8) || (X == Y))

- ⑤ ((X == 3) && ((Y < 0) || (Z < 5)))

- ⑥ ((!(X < 10) || (Y > -5)) && (Z > 8))

- ⑦ ((X > 0) && (Y < 0) && (Z > 0) && ((X + Z + Y) > 0))

- ⑧ (((X > 0) && !(X < 10)) || ((Y >= -2) && (Z >= 8)) && (X > Y))

Lesson4 算術・比較演算の優先順位



Point◆◇優先順位を明確化しよう!!

C言語の全ての演算子は処理される優先順位が決まっています。
複数の演算子を使用した処理には、適切な場所に括弧を付けることにより処理の順序が明確化されます。

【問題1】 次の式に括弧()を付けて演算順序を明確にしてください。

- ① 時分秒の時間が入力されている変数 x(時)、y(分)、z(秒)から秒を求める。

演算式 $sec = x * 3600 + y * 60 + z;$

- ② 体温が入力されている変数 x (°C)が 37°C以上 39°C未満か判定する。

判定式 ($x \geq 37 \ \&\& \ x < 39$)

- ③ 車速が入力されている変数 x (km/h)が 100km/h 以上 150km/h 未満、又はエンジン回転数が入力されている変数 y (rpm)が 3000rpm 以上 5000rpm 未満か判定する。

判定式 ($x \geq 100 \ \&\& \ x < 150 \ || \ y \geq 3000 \ \&\& \ y < 5000$)

- ④ getch 関数で入力された変数 x の文字が a か判定する。

判定式 ($x = \text{getchar}() == 'a'$)

Lesson5 if文



Point◆◇条件分岐を理解しよう!!

if文は、制御の流れを2方向に変える命令で、分岐する方向は条件式によって決定されます。
条件式は「|| (or)」、「&& (and)」を使い複数組み合わせることが可能です。

【問題1】 次の□部分を解答群から選び、プログラムを完成させなさい。

【処理内容】

入力された変数 x が 5 以上のとき変数 y に 1 を、それ以外のときは変数 y に 3 を代入する。

```
void main(void)
{
    int x;
    int y;

    y = 0;
    printf("数値入力:");
    scanf("%d",&x);
    if( □① )
    {
        y = 1;
    }
    □②
    {
        y = 3;
    }
    printf("%d¥n",y);
}
```

解答群

ア. x > 5
イ. x < 5
ウ. x >= 5
エ. else if
オ. else
カ. x == 5

【問題2】 次の□部分を解答群から選び、プログラムを完成させなさい。

【処理内容】

代入された変数 x が 10 以下のとき変数 y に 1 を、変数 x が 20 以上のとき変数 y に 3 を、
それ以外のときに変数 y に 6 を代入する。

```
void main(void)
{
    int x;
    int y;

    y = 0;
    printf("数値入力:");
    scanf("%d",&x);

    if( □① )
    {
        y = 1;
    }
    else if( □② )
    {
        y = 3;
    }
    else
    {
        □③ ;
    }
    printf("%d¥n",y);
}
```

解答群

ア. x > 10
イ. x < 10
ウ. x <= 10
エ. x < 20
オ. x > 20
カ. x >= 20
キ. y == 6
ク. y = 6
ケ. y != 6

【問題3】 次の□部分を解答群から選び、プログラムを完成させなさい。

【処理内容】

入力された変数 x が 0 以上かつ 10 以下のとき、変数 x を 2 で割った余りにより奇数か偶数か判定する。

※ただし、入力された変数 x が上記以外のときはエラーを表示する。

```
void main(void)
{
    int x;

    printf("入力:");
    scanf("%d",&x);

    if(( □① ) □② ( □③ ))
    {
        if((x □④ 2) == 1)
        {
            printf("奇数です\n");
        }
        else
        {
            printf("偶数です\n");
        }
    }
    else
    {
        printf("エラー\n");
    }
}
```

解答群

ア. $x \geq 0$
イ. $x < 0$
ウ. $x > 0$
エ. $ $
オ. $\&\&$
カ. $==$
キ. $x < 10$
ク. $x > 10$
ケ. $x \leq 10$
コ. $\%$
サ. $/$
シ. $+$

【問題4】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

入力された演算子に応じて、 x と y の計算を行い、結果を表示する。

※ただし、入力する演算子は、 $+$ 、 $-$ 、 $*$ 、 $/$ のいずれかとし、それ以外が入力された場合は"エラー"を表示する。

```
void main(void)
{
    char ch;
    int x = 10;
    int y = 2;
    int z = 0;

    printf("記号入力:");
    scanf("%c",&ch);

    if( ch == □① )
    {
        z = x + y;
        printf("加算結果:%d\n",z);
    }
    else if( ch == □② )
    {
        z = x - y;
        printf("減算結果:%d\n",z);
    }
    else if( ch == □③ )
    {
        z = x * y;
        printf("乗算結果:%d\n",z);
    }
    else if( ch == □④ )
    {
        z = x / y;
        printf("除算結果:%d\n",z);
    }
    else
    {
        printf("エラー\n");
    }
}
```

Lesson6 switch 文



Point◆◇切り替えを理解しよう!!

コンピュータの処理においては、条件により様々な処理へ分岐するものもあります。
ここでは複数の分岐へと分かれる多分岐型の処理について覚えてください。

【問題1】 次の□部分を解答群から選び、プログラムを完成させなさい。

【処理内容】

変数 x に入力された値によって変数 y への加算する値を変更する。
入力値が 1 のとき 3 を、入力値が 2 のとき 2 を、入力値が 3 のとき 1 を変数 y に加算する。
それ以外の値のときは変数 y に 0 を代入する。

```
void main(void)
{
    int x;
    int y = 10;

    printf("入力:");
    scanf("%d",&x);

    switch( □ ① □ )
    {
        case 1: □ ② □ y += 3;
        case 2: □ ② □ y += 2;
        case 3: □ ② □ y += 1;

        □ ③ □ : y = 0;
        □ ② □ ;
    }
    printf("結果:%d\n",y);
}
```

解答群

ア. x
イ. y
ウ. return
エ. break
オ. else
カ. default

【問題2】 次の□部分を解答群から選び、プログラムを完成させなさい。

【処理内容】

変数 code に入力された値に対応した年号を表示する。年号に対応する値は、明治:1、大正:2、昭和:3、平成:4 で、それ以外の値が入力された場合は"エラー"を表示する。

```
void main(void)
{
    char code;

    printf("入力:");
    scanf("%d",&code);
    switch( □ ① □ )
    {
        □ ② □ 1: printf("年号:明治\n");
        break;
        □ ② □ 2: printf("年号:大正\n");
        break;
        □ ② □ 3: printf("年号:昭和\n");
        break;
        □ ② □ 4: printf("年号:平成\n");
        break;
        □ ③ □ : printf("エラー\n");
        break;
    }
}
```

解答群

ア. break
イ. default
ウ. case
エ. if
オ. else if
カ. else
キ. code

【問題3】 以下のプログラムを実行し、変数 x に 1 を入力したときの変数 y の値を答えなさい。

```
void main(void)
{
    int x;
    int y;

    y = 0;
    printf("x 入力:");
    scanf("%d",&x);

    switch(x)
    {
        case 1:
            y = y + 1;
        case 2:
            y = y + 2;
        case 3:
            y = y + 3;
        break;
    }
    printf("y 出力:%d¥n",y);
}
```

【問題4】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

変数 x 、変数 y に入力された値が両方とも奇数か、片方が奇数か、両方とも偶数かを判断し出力する。

```
void main(void)
{
    int x,y;
    int cnt = 0;

    printf("x 入力:");
    scanf("%d",&x);
    printf("y 入力:");
    scanf("%d",&y);

    if( ( □①□ ) == 1)
    {
        cnt += 1;
    }
    if( ( □②□ ) == 1)
    {
        cnt += 1;
    }
    switch( cnt )
    {
        □③□
            printf("両方とも偶数¥n");
            break;
        □④□
            printf("片方が奇数¥n");
            break;
        □⑤□
            printf("両方とも奇数¥n");
            break;
    }
}
```

Lesson7 for 文



Point◆◇ループの特性を理解しよう!!

コンピュータの処理において、条件によって同じ処理を繰り返す反復型の処理があります。
ここでは規定回数処理を繰り返す for 文について練習しましょう。

【問題1】 次の□部分を解答群から選び、プログラムを完成させなさい。

【処理内容】

変数 k に入力された値の回数、"1 2 3 "を繰り返し表示する。

```
void main(void)
{
    int i,j,k;

    printf("入力:");
    scanf("%d",&k);

    for( □①□ ; □②□ ; □③□ )
    {
        for(j = 1; j <= 3 ; j++)
        {
            printf("%d ",j);
        }
        printf("¥n");
    }
    printf("¥n");
}
```

解答群

ア. i <= 0
イ. i = 0
ウ. i == 0
エ. i <= k
オ. i > k
カ. i < k
キ. k++
ク. j++
ケ. i++

【問題2】 次の□を埋め、プログラムを完成させなさい。

【処理内容】

変数 i が 0 から 5 まで増加し、i の各値ごとに変数 j が 1 から 3 まで変化する。変数 k に変数 i と変数 j の値を加算し続ける。

```
void main(void)
{
    int i;
    int j;
    int k;

    k = 0;
    for( □①□ ; □②□ ; i++ )
    {
        for( □③□ ; □④□ ; j++ )
        {
            □⑤□ ;
        }
    }
    printf("%d¥n",k);
}
```

【問題3】 次の□を埋め、プログラムを完成させなさい。

【処理内容】

変数 i が 0 から 10 まで 1 ずつ増加する間、変数 i を 2 で割った剰余が 1 のとき変数 i を変数 sum に加算する。

```
void main(void)
{
    int i;
    int sum;

    sum = 0;
    for( □①□ ; □②□ ; □③□ )
    {
        if( □④□ == 1)
        {
            sum = sum + i;
        }
    }
    printf("奇数合計:%d¥n", sum);
}
```

【問題4】 以下のプログラムを実行した場合、どのような出力結果となるか書きなさい。

```
void main(void)
{
    int i;
    int j;

    for( i = 0 ; i < 3 ; i++ )
    {
        for( j = 0 ; j < 4 ; j++ )
        {
            printf("*");
        }
        printf("¥n");
    }
}
```

Lesson8 while 文



Point◆◇ループの特性を理解しよう!!

コンピュータの処理においては、終了条件を指定し、その間処理を繰り返す反復型の処理があります。
ここでは条件を満たすまで処理を繰り返す while 文について覚えてください。

【問題1】 次の□部分を解答群から選び、プログラムを完成させなさい。

【処理内容】

変数 no が 10 を超えるまで 3 を加算し続ける。

```
void main(void)
{
    int no;

    no = 0;
    while( □① )
    {
        no += 3;
    }
    printf("合計:%d¥n",no);
}
```

解答群

ア. no != 10
イ. no = 10
ウ. no == 10
エ. no <= 10
オ. no > 10

【問題2】 次の□部分を解答群から選び、プログラムを完成させなさい。

【処理内容】

入力された数値を表示する。但し 99 が入力された場合は表示後にプログラムを終了する。

```
void main(void)
{
    int i;

    □①
    {
        printf("数値入力:");
        scanf("%d",&i);
        printf("入力表示:%d¥n",i);
    }while( □② );
}
```

解答群

ア. go
イ. start
ウ. do
エ. i != 99
オ. i = 99
カ. i == 99

【問題3】 次の□部分を埋め、以下の処理結果になるようにプログラムを完成させなさい。

【処理結果】

```
****
****
****
```

```
void main(void)
{
    int i;
    int j;

    i = 1;
    while( □① )
    {
        j = 1;
        while( □② )
        {
            printf("*");
            j++;
        }
        printf("%n");
        i++;
    }
}
```

【問題4】 次の□部分を埋め、プログラムを完成させなさい。

【処理内容】

入力された変数 cnt が 0 より大きい整数の場合、1 から cnt まで順に表示する。

```
void main(void)
{
    int cnt;
    int i;

    printf("入力:");
    scanf("%d",&cnt);
    if( □① )
    {
        printf("出力:");
        i = 1;
        while( □② )
        {
            printf("%d ",i);
            i++;
        }
    }
}
```


解答

Training4 判断と制御

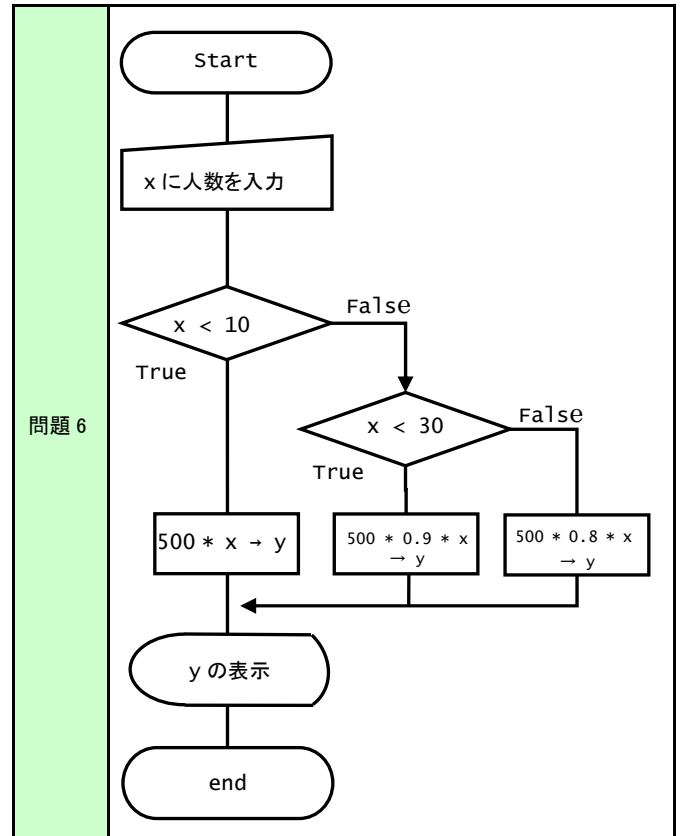
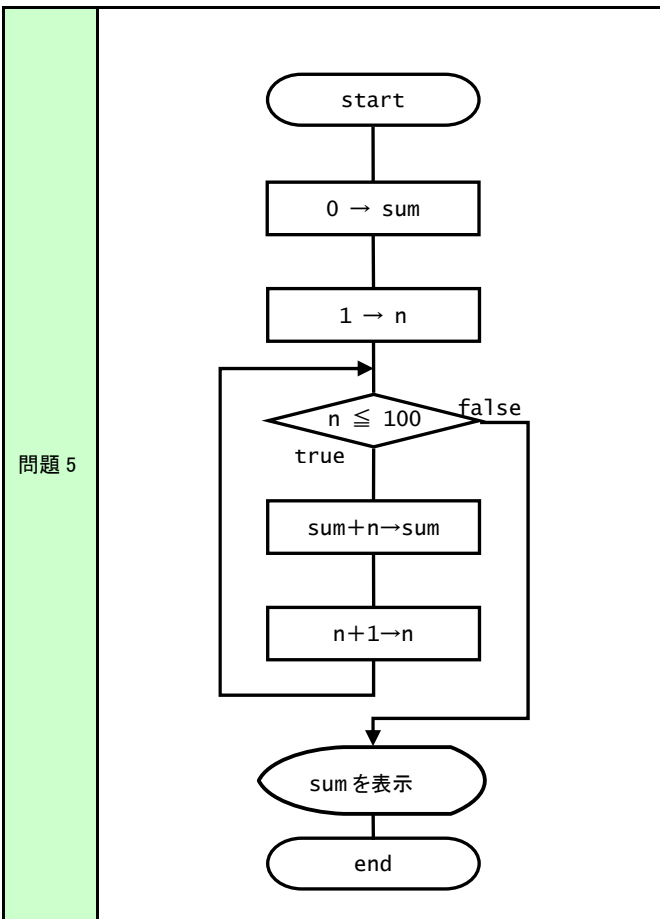
Lesson1 フローチャート

問題 1 ①カ ②オ ③エ ④ア ⑤イ

問題 2 ①イ ②ウ ③ア ④エ

問題 3 ①エ ②ア ③ウ ④イ

問題 4 ①イ ②カ ③ウ ④オ ⑤ア



【解説】問題 6

この解答はあくまで 1 つの例である。
以下の要点が正しく書かれていれば正解とする。

- ・ start の端子で開始し、end の端子で終了する。「開始」「終了」などといった、意味が同じである言葉を使用してもよい
- ・ 人数を入力するための記号がある。
- ・ 条件式が正しく書かれている。($x \geq 30$, $x \geq 10$ と書いても、その後続く料金を計算する式が対応していればよい)
- ・ 条件式の記号の横または下には True , False といった真偽をあらわす言葉が書かれている。
- ・ 入園料金の合計を表示する記号がある。

Lesson2 比較演算

問題 1 ①>= ②<= ③< ④> ⑤== ⑥!=

問題 2 ①偽 ②真 ③真 ④偽 ⑤偽 ⑥真
⑦真 ⑧偽

【解説】問題 2

④演算子の優先順位により、割り算 ($y / x = 2$) から行われます。2>3 の結果、偽となります。

⑤演算子の優先順位により、 $y < x$ から行われます。y の方が大きいのでこの結果は偽となり、0 が返されます。

そして、1(z) と 0(偽) の比較の結果、偽となります。

⑧演算子の優先順位により、 $x > y$ 、 $x < z$ をそれぞれ行います。その結果、 $x > y$ は偽となり 0 が返され、 $y < z$ も偽なので 0 が返されます。そして $0 != 0$ の結果、偽となります。(0 と 0 で、等しくない(!=)か比べている)

Lesson3 論理関係演算

問題 1 ①&& ②|| ③! ④&&, ||

問題 2 ①偽 ②真 ③偽 ④真
⑤真 ⑥偽 ⑦真 ⑧真

Lesson4 算術・比較演算の優先順位

問題 1 ①sec = (x * 3600) + (y * 60) + z ;
②((x >= 37) && (x < 39))
③(((x >= 100) && (x < 150)) ||
((y >= 3000) && (y < 5000)))
④((x = getchar()) == 'a')

Lesson5 if 文

問題 1 ①ウ ②オ

問題 2 ①ウ ②カ ③ク

問題 3 ①ア ②オ ③ケ ④コ

問題 4 ①'+ ' ②'- ' ③'* ' ④'/'

Lesson6 switch 文

問題 1 ①ア ②エ ③カ

問題 2 ①キ ②ウ ③イ

問題 3 6

問題 4 ①x % 2 ②y % 2 ② case 0 :
③ case 1 : ④ case 2 :

Lesson7 for 文

問題 1 ①イ ②カ ③ケ

問題 2 ①i = 0 ②i <= 5 ③j = 1 ④j <= 3
⑤k = k + i + j ;

問題 3 ①i <= 0 ②i <= 10 ③i++ ④(i % 2)

問題 4 ****

Lesson8 while 文

問題 1 エ

問題 2 ①ウ ②エ





問題 3 ①i <= 3 ②j <= 5

問題 4 ①cnt > 0 ②i <= cnt

巻末資料

フローチャートの書き方

記号の説明

			
処理内容を示す	分岐するための条件を示す	流れの始めと終わりを示す	ループの始端と終端を示す

【代入処理】

「 $10 \rightarrow n$ 」の記述は 10 を n に代入という意味である。
また、「 $n + 1 \rightarrow n$ 」は $n + 1$ を n に代入という意味になる。

【分岐条件】

条件に対して真と偽で分岐する。
「 $n > 100$ 」が分岐条件の場合、 $n > 100$ なら真、そうでないなら偽になる。

【ループ処理】

ループ端で繰り返す処理内容を挟む。ループ始端にループの継続条件(終了条件)を書く。
※「 $n \leq 5$ 」の場合、 n が 5 以下の時繰り返す。
※「 $i : 0, 1, 5$ 」がループ条件の場合、 i の初期値は 0 であり、ループ始端からループ終端まで実行するごとに i を 1 加算。
5 になるまで繰り返す。