Vol.2011-CE-109 No.7
2011/3/18

文系学部における PEN を用いたプログラミング授業の実践例

ー繰り返し処理の理解を助ける教材の提案ー

松本このみ† 吉田智子†

文系学部のみを持つ京都ノートルダム女子大学では，全学共通の情報リテラシー科目「情報処理」(90 分/コマ×15 コマ)の中で，2007 年度より，PEN を用いたプログラミング導入授業を 4 コマ分，実施している．2009 年度までの授業において，繰り返し処理が理解できない学生が多かった．そこで 2010 年度の授業では，文系学部の学生用に工夫した「繰り返し処理の理解を助ける教材」を作成し，授業で利用した．この教材内容と授業実践による考察を紹介したい．

Programming Education for Liberal Arts College Students with PEN

ーProposal of New Materials for Teaching Loop Structureー

Konomi Matsumoto† and Tomoko Yoshida†

At Kyoto Notre Dame University, we included programming education into "Information Processing" class for liberal arts students by using "PEN:Programming Environment for Novices" in 2007. But it has been difficult to make students understand loop structure programs. We introduced new materials for teaching loop structure. In this paper, we report on our experiences and student responses.

1. はじめに

高校での教科「情報」の必修化に伴い，2006 年 4 月以降の大学への入学者は，高等学校での教科「情報」を履修した学生となった．しかし，高校での「情報」の授業は，ワープロや表計算の使い方を学ぶケースが多い．大岩は，情報教育の目的は，コンピュータが利用できることと同時に，コンピュータの可能性と限界を理解することだと主張している [1]．また文献 2)では，社会に普及している情報システムの特性を理解するためには，プログラミングなどを通して「コンピュータによる自動的な処理」に接した体験を持つべきだと述べている．

この考えを受け，文系学部のみを持つ女子大学である本学においても，情報教育の中でプログラミングを学ぶ授業を，2005 年度より学科選択の専門科目の中で，2007 年度からは全学共通の教養科目である「情報処理」の中で実施している [3]．しかし，「情報処理」授業において，繰り返し処理が理解できない学生が多かったため，2010 年度の授業においては，「繰り返し処理の理解を助ける教材」を作成し，それを授業で利用した．そこで今回の報告では，その教材の内容と授業実践による考察を紹介したい．

2. 本学の情報教育の概要とプログラミング導入の問題点

2.1 本学の情報教育への PEN の導入理由

本学において，筆者らが担当するプログラミングの授業で利用している環境が，PEN(Programming Environment for Novices)である．PEN は，西田ら [4]によって開発された，初学者向けのプログラミング環境で，初めてプログラミングを経験する学生にも馴染みやすい機能が多く含まれている．代表的なものが，プログラムが日本語で記述できることや，ボタンで構文を入力できることや，実行中の変数の値が画面に表示されることである [5].

本学で開講されている全学共通科目の情報関連科目は，1 年生の必修科目の「情報演習Ⅰ（半期，90 分×15 コマ)」，選択科目の「情報演習Ⅱ（半期，90 分×15 コマ)」と「情報処理（半期，90 分×15 コマ)」の 3 科目である．「情報演習Ⅰ」と「情報演習Ⅱ」は，MOS(Microsoft Office Specialist)などの資格に対応する授業内容となっており，Windows 上のオフィスツールを中心に扱った授業である．

一方の「情報処理」では，筆者が 1998 年に授業を開始して以来，ネットワークリテラシーを身につけるための教育内容を核としてきた [6]．Windows 上のオフィスツールの説明は，別に必修科目が存在するため，教える側が「コンピュータの可能性と限

† 京都ノートルダム女子大学 人間文化学科
Department of Cross-Cultural Studies, Kyoto Notre Dame University

界を理解させるために必要だ」と思う教育を実施することが可能であった [7]．そのため，2000 年度より，UNIX 系 OS を利用させる内容を含めたが，そこにさらに，2007 年度より，PEN によるプログラミング体験を含めた．通常のプログラミング言語では，4 コマで学生にプログラミングを教えることは難しいが，PEN ならそれが可能ではないかと考えた．

2.2 2009 年度までのプログラミング授業の問題点

2009 年度までの授業では，逐次処理，条件分岐までは理解できる学生が多いが，繰り返し処理の理解度が低いことが明らかになった [3]．その原因の一つが，利用テキストの繰り返し処理の部分で扱っている問題が，全体的に数値（倍数や階乗）を扱うもので，本学の学生にとって，面白さに欠けるものであることが考えられた．

本学の学生は，コンピュータで絵を描くような操作を好む傾向が強い．例えば，「PEN の図形描画の機能を使って，自由に絵を描きなさい」という時間を与えると，コツコツと座標値を入力していく．授業中に時間が足りなければ，自習時間を使ってでも，教師側が感動するような絵を仕上げるのである．2007 年度の授業では，32 作品のうち 15 作品が逐次処理しか使っていなかったが，それらの多くは独自性に溢れる，素晴らしい作品であった [3]．つまり，自分が興味のあることであれば集中して取り組むが，それ以外のことには，拒否反応を示す傾向がみられた．

このような背景からも，本学のプログラミング導入において，特に繰り返し処理の理解を高めるためには，学生が興味を持って取り組める教材を与える必要があると考えた．

3. PEN による 4 コマの授業内容（2009 年度まで）

以下に，2009 年度までの演習内容を紹介する．各回とも，例題をもとに解説した後，練習問題（付録 1）をやらせた．

・ 第 1 回　PEN の概要・逐次処理

最初の授業では，プログラミングを学ぶ意義，PEN の使い方，変数の概念や制御構造等について説明した．次に，逐次処理の例題を利用し，変数の宣言，代入文，出力文の使い方を解説した．PEN の一行実行の機能についても紹介した．

例題 1.1： キーボードから 1 つの整数を入力し，それを 3 倍した値を出力するプログラム
例題 1.2： キーボードから長方形の二辺の長さを入力し，面積を表示するプログラム

・ 第 2 回　条件分岐

条件分岐について，条件式に使う不等号や，論理演算子などの説明をした．

例題 2.1： キーボードから入力した数が，3 で割り切れるかどうかを調べ，割り切れるなら「3 の倍数である」と出力するプログラム
例題 2.2： キーボードから入力した数が，3 で割り切れるかどうかを調べ，割り切れるなら「3 の倍数」と出力し，そうでないなら「3 の倍数でない」と出力するプログラム
例題 2.3： 入力された数（点数） が 90 以上なら「A」，80 以上 90 未満なら「B」，70 以上 80 未満なら「C」，60 以上 70 未満なら「D」，60 未満なら「不合格」と出力するプログラム

・ 第 3 回　繰り返し処理

繰り返し処理について説明した後，例題を用いて，繰り返し処理の中ではどのようにプログラムが実行され，制御変数がどう変わっていくかを解説した．

例題 3.1： 10 から 20 までのそれぞれの数の 2 乗を求めるプログラム
例題 3.2： 一つの小さな整数 n を入力しその階乗を求めるプログラム

・ 第 4 回　図形描画

図形描画を行う関数の使い方を説明した後，例題を使って，そのプログラムの中身を修正したり書き加えたりして，どのような関数を使えば，どのように絵が変わっていくのかを解説した．

例題 4.1：ウィンドウに円や長方形などの基本的な図形を描くプログラム
例題 4.2：RGB 値を画面から入力させて，毎回，違う色の図形を描くプログラム
例題 4.3：400×400 の描画ウインドウに，半径 25 の赤い円を互いに接するように水平に配置するプログラム
例題 4.4：お花の描画プログラム（花びらのサイズを画面から入力）

4. 2010 年度の繰り返し処理の教材

2009 年度までの繰り返し処理の教材の問題点と改善提案を挙げると共に，筆者らが 2010 年度の授業のために作成した教材を紹介する．

2010 年度も「情報処理」の 4 コマをプログラミングのために使い，第 3 回の繰り返

し処理における教材の内容を変更した以外は，2009 年度までと基本的に同じ授業内容である．ただし，繰り返し処理における教材の変更に伴って，2 回目の授業内容を少し変更している（詳しくは，4.2.2 参照）．

4.1 2009 年度までの教材の問題点

2009 年度までの教材では，以下の点が，繰り返し処理の理解を困難にしているのではないかと考えた．

(1) 学生にとって，例題および練習問題のテーマに興味が持てないこと
(2) 制御変数の使い方が理解できないこと
(3) 問題ごとに変数名や変数の役割が異なっていたこと

そこで，(1)に関しては，本学の学生にとって，テーマに馴染みのある問題を用意した．(2)に関しては，制御変数を使う必要のない，単純な繰り返しのプログラムから始めるのがよいのではないかと考えた．(3)に関しては，例題や練習問題に，新規のテーマのプログラム作成問題をいくつか用意するのではなく，学生がすでに理解できているプログラムを，少しずつ書き直すことで，プログラミングの楽しさを実感させる方法がよいと考えた．

4.2 繰り返し処理の理解を助ける教材の提案

ここで，実際に 2010 年度の 3 コマ目の繰り返し処理の授業で使った教材を紹介する．まず，繰り返し処理の導入となる前半には，次のような例題と練習問題を用意した．

・ 2010 年度授業の第 3 回　繰り返し処理

例題 3.1： いちご（単数形）の正しい英語のスペルを入力させるプログラム
例題 3.2： スーパーで，20 個，買い物をしました．その日は特別に，それぞれ 20 円ずつ値引きしてもらえました．消費税は考えないものとして，買ったものの金額を一つずつ入れていくと，値引き後の合計金額が表示されるプログラムを書きなさい．
練習問題 3.1： 例題 3.2 を応用し，購入した品物の数を入力し，合計金額を求めるプログラム
練習問題 3.2： 練習問題 3.1 のプログラムを応用し，100 円以上の品物の場合だけ 20 円引きして，合計金額を求めるプログラム

授業の後半では，2009 年度までに実施していた例題や練習問題の中から，次のものを紹介して，学生に実行させた．

例題 3.3： 10 から 20 までのそれぞれの 2 乗を求めるプログラム
例題 3.4： 1 から 10 までの整数の和を求めるプログラム

以下に，例題 3.1 と例題 3.2 について詳しく説明する．

4.2.1 例題 3.1：「いちご問題」

例題 3.1：いちご（単数形）の正しい英語のスペルを入力させるプログラム

```
1: 整数 a
2: 文字列 word
3: a ← 1
4: 「いちご（単数形）のスペルを入力してね：」 を改行なしで表示する
5: word ← input()
6: word ≠ 「strawberry」 の間，
7:     | a ← a + 1
8:     | 「スペルが間違っています．もう一度入力してね：」 を表示する
9:     | word ← input()
10: を繰り返す
11: 「正解です．いちごのスペルは」 と word と 「です.」 を表示する
12: 「あなたは」 と a と 「回目の挑戦で正解しましたね.」 を表示する
```

2009 年度までの授業では，繰り返し処理の導入時に最初に示す例題が，「2 乗を求めるプログラム」であったため，4.1 で述べた (1)と(2)の問題が生じた．一方，この「いちご問題」では，繰り返し処理の中に制御変数を含めていない．それに加えて，ゲーム感覚で楽しめる，誰にでも馴染みやすいテーマとなっている．

4.2.2 例題 3.2：「買い物問題」

次に，学生がすでに理解できているプログラムを，段階を踏んで少しずつ書き直していくことでハードルを低くし，最終的に自分でプログラミングを行なうことの楽しさを実感させるために考えた例題を紹介する．この例題は，4.1 で述べた(1)と(3)の問題の解決を意図している．

例題 3.2：スーパーで，20 個，買い物をしました．その日は特別に，それぞれ 20 円ずつ値引きしてもらえました．消費税は考えないものとして，買ったものの金額を一つずつ入れていくと，値引き後の合計金額が表示されるプログラムを書きなさい．

まず，「スーパーでの買い物」というテーマは，「いちご問題」と同様に，学生が身近に感じるものである．また，第 3 回の授業でこの例題を取り上げることを見越して，2010 年度の第 2 回の授業では，次のような，逐次処理を使った「2 個の商品の合計金額を求めるプログラム」を，練習問題として取り組ませていた．

```
1: 整数 a, b
2: 整数 gokei
3: 「一つ目の買い物の金額を入力してね：」を表示する
4: a ← input()
5: 「二つ目の買い物の金額を入力してね：」を表示する
6: b ← input()
7: gokei ← (a - 20) + (b - 20)
8: 「それぞれ 20 円割引した合計金額は」を改行なしで表示する
9: gokei を改行なしで表示する
10:「円です.」を表示する
```

このプログラムは，逐次処理だけを使ったものであるため，学生はスムーズに理解できている．最初に，この逐次処理のプログラムを学生に PEN で実行させて，それぞれの変数の役割を思い出させる．

次に，20 個の商品の合計金額を求めるプログラムに変更する方法を考えさせ，講師が書いた「買い物したそれぞれの金額を，a から t までの変数に格納するプログラム例（46 行）」を見せた．

学生が「買い物する商品の数の分，変数を用意するのは大変だ」と感じた後で，「買い物した商品の値段をその都度，a に入れて，gokei に足し込んでいく処理」を 20 回記述して，20 個の商品の合計金額を計算するプログラム例を示した．使う変数の数は少なくなったが，プログラムは 65 行もの長いものであることを，学生に視覚的に示した．

最後に，繰り返し処理を使うことで， 12 行のプログラムが書けることを示し，学生には，「2 個の商品の合計金額を求める」プログラムを「繰り返し処理を使って 20 個の合計金額を求める」ものに変更させ，PEN で実行させた．以下が，そのプログラムである．

```
1: 整数 a
2: 整数 gokei, counter
3: counter ← 1
4: counter ≦ 20 の間,
5:       | 「買い物の金額を入力してね：」を表示する
6:       | a ← input()
7:       | gokei ← gokei + (a - 20)
8:       | counter ← counter + 1
9: を繰り返す
10:「それぞれ 20 円引きした合計金額は」を改行なしで表示する
11: gokei を改行なしで表示する
12: 「円です.」を表示する
```

以上のように段階を踏んで説明することで，繰り返し処理や制御変数の必要性をスムーズに理解させることができた．

その後，学生には「練習問題 3.1（前述）」として，例題 3.2 のプログラムに機能を追加する形で，購入した品物の数を入力し，合計金額を求めるプログラムと，「練習問題 3.2（前述）」として，100 円以上の品物の場合だけ 20 円引きして，合計金額を求めるプログラムを書かせた．

そして，授業の後半で，2009 年度までに実施していた例題や練習問題の一部（前述）をフローチャート図を加えるなどの工夫をして提供した．

5. 考察

前節で紹介したように，2010 年度の繰り返し処理の授業では，導入問題に制御変数を使わない繰り返し処理の例（「いちご問題」）を用いたり，例題のテーマを身近なもの（「買い物問題」）に変更した．さらに，繰り返し処理の必要性を明確にし，思考の流れに沿って段階的にプログラムを変更していくことで，プログラミングの初学者には理解しづらい，繰り返し処理の理解を助ける教材を提供した．

2010 年度の授業を受けた学生の理解度を 2007 年度と比較してみた．2007 年に行った調査 [3] と同じ復習問題を使用した（図 1，参照）．問題の変更点は変数名を一部馴染みやすいものにしただけで，選択肢は全て同じである．この問題の結果は以下のようになった（図 2）．

Vol.2011-CE-109 No.7
2011/3/18

復習問題

以下のプログラムは、キーボードから 10 人分の点数（0～100 までの整数）を入力し、最高点と最低点を見つけて表示するためのプログラムである。空欄（あ）～（お）に入れるのに適当な語句・数を解答群の中から選び、その記号を記せ。

```
1:  整数 a, b, counter, x
2:  a ← input()     /* 最初の数を読み込む */
3:  b ← a
4:  counter ← 1
5:  counter < （あ）の間，
6:      | x ← input()
7:      | もし x < b ならば
8:      |   | b ← x
9:      | を実行する
10:     | もし （い） ならば
11:     |   | a ← x
12:     | を実行する
13:     | (う)
14: を繰り返す
15: 「最高点は：」と (え) と「、最低点は：」と (お) を表示する
```

[解答群]

(a) -1	(b) 0	(c) 1
(d) 2	(e) 9	(f) 10
(g) a	(h) b	(i) x
(j) counter	(k) a ← a + 1	(l) x ← x + 1
(m) counter ← counter + 1	(n) counter + 1	(o) x + 1
(p) x > b	(q) x < a	(r) x > a

図 1　穴埋めの復習問題

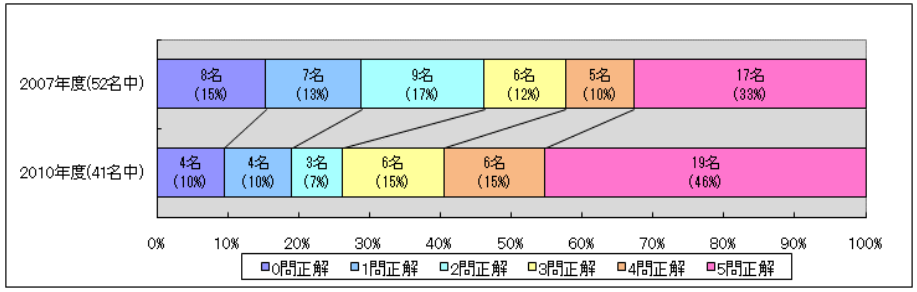

図 2　2007 年度と 2010 年度の復習問題の正解率比較

この結果から，2007 年度に比べて 2010 年度の正解者の割合が増えていることがわかる．

また，2010 年度のプログラミングの授業中の学生の様子を 2009 年度の様子と比べてみると，自分自身でプログラムを考えて入力するという作業を楽しむ学生が多くなったという印象がある．そこで，PEN が自動的に生成する各学生の実習のログ（入力したプログラムの全行，画面入力内容，コンパイル時のエラーログなどが一つのファイルとして保存される）を眺めてみたところ，ログが 2KB 以下の学生は，サンプルプログラムを入力しただけ，もしくはコンパイルの回数が極端に少ないという傾向がみられた．一方，ログのサイズが大きい学生は，何度もコンパイルを繰り返し，課題に挑戦していることがうかがえた．そこで，各学生のログのサイズを，2009 年度と 2010 年度とで比較してみた．

比較の対象としたのは，第 3 回の繰り返しの授業を受けた，2009 年度の学生 34 名と，2010 年度の学生 40 名のログである．おおまかな判断ではあるが，ログが 2KB 以下の学生は教員が示したサンプルプログラムを入力しただけであるため,「消極的学習者」とした．そして，ログが 4KB 以上の学生を，自分でプログラムを考えて入力した「積極的学習者」と判断して，比較した（図 3，参照）．

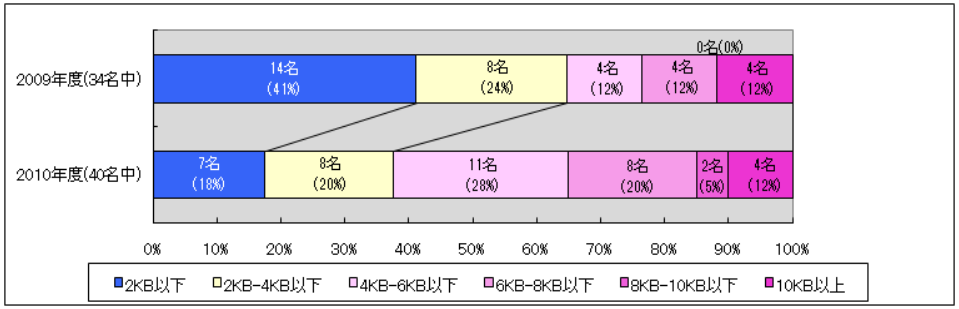

図 3　2009 年度と 2010 年度の PEN のログのサイズ比較

このグラフからわかるように，2009 年度は 34 名中 14 名（41.1%）だった消極的学習者の数が，2010 年度には 40 名中 7 名（17.5%）と，大幅に減っていることがわかる．また，ログが 4KB 以上の積極的学習者の割合は，2009 年度は，34 名中 12 名（35.3%）だったのが，2010 年度には 40 名中 25 名（62.5%）と，半数以上に増えている．この数字は，第三回の繰り返しの授業中に，講師側から見て，つまらなそうにしていた学生の割合と，楽しそうに課題に挑戦していた学生の割合に，ほぼ等しいと思われる．

さらに，このプログラミング体験から，プログラミングが面白いと感じた学生の割合を比較してみる．2007 年度では，プログラミングを面白いと感じた学生は，52 名中 29 名（56%）であったが，2010 年度では 49 名中 30 名（61%）であった．

2010 年度の調査で，「面白かった/面白くなかった」と感じる学生の感想は，それぞれ次の通りである．

【面白かった】
・難しかったけど，出来たときの達成感が最高だったし，入力するとすぐに答えがでて，とても簡単で便利だと感じた．また PEN を使って授業がしたい
・プログラミングというものをやったことがなかったので，興味深く，実際にプログラムを作り上げていく過程が面白かったです
・最初は全くわからなかったが，4 コマ受けて少しずつ出来るようになった．とくに，描画関数が面白かった

【面白くない】
・難しかったから
・算数的な思考がなくて苦手だったから．むずかしかった

このように，面白くないと答えた学生はプログラミング自体が難しいと感じていたり，進むのが速いと感じていたり，また数学的な部分が難しいと感じているようである．面白くないと答えた学生も，否定的な意見ばかりでなく少なからず達成感を覚えたという意見もいくつかあった．以下にいくつかその感想を挙げておく．

【面白くない（けれども楽しめた）】
・実行できたときは嬉しかったが，授業のスピードが速すぎてついていくのが大変だった
・とても難しかったから．でも，実行したプログラムが，スムーズに動いたときは気持ちよかった

6. おわりに

今回の試みは，2009 年度までのプログラミングの授業内容を，2010 年度に文系の学生向きに変えることで，学生の理解度を高めるということを目的としていた．そこで実際に繰り返し処理に関する教材を作成し，それを「情報処理」の授業で使用した．その結果，復習問題の正答率が高くなったり，積極的学習者の割合が増えたり，プログラミングの体験が面白かったと答えた割合が増えた．プログラミング初学者向けの繰り返し処理の導入教材が用意できたことは，一つの成果だと考える．

しかし，反省点もいくつか挙げられる．例えば，最初の導入問題で紹介する「いちご問題」では，何度入力したかを数える変数を使わなければ，よりシンプルなプログラムが提供できた．次年度からは，この，よりシンプルな問題を例題として始めに提供し，その応用問題として，何度入力したかを数える変数を追加した練習問題を与える予定である．

また，今回「買い物問題」の後に提供した練習問題の二問は，例題のプログラムに，追加機能を加えるだけのもので，繰り返し処理の部分を新たに書く必要がなかった．しかし，せっかく繰り返し処理を理解した後なので，「20 人の学生のテストの点数を画面から入力させて，平均点を求める」といった問題を与える必要もあっただろうと考える．

今回うまくいった点は次回にも活かせるようにし，また反省点は次回には改善出来るようにして，今後もよりよい授業が展開出来る教材を考えていきたい．

謝辞 この研究および本稿の執筆に関して，PEN の開発者の一人である，大阪市立大学の松浦敏雄教授にお世話になった．この場を借りて，お礼を申し上げたい．

参考文献

1) 大岩元:「高校における教科情報としてのプログラミング教育」,『コンピュータと教育 40-8』, 情報処理学会，pp.53-60(1996.5).
2) 情報処理学会情報処理教育委員会："日本の情報教育・情報処理教育に関する提言 2005"
3) 吉田智子:「文系学部の情報教育へのプログラミングの導入～PEN を用いた実践例～」,『コンピュータと教育』情報処理学会研究報告，2008-CE-95(12)，pp.71-48(2008.5).
4) 西田 知博，原田 章，中村 亮太，宮本 友介，松浦 敏雄:「初学者用プログラミング学習環境 PEN の実装と評価」，情報処理学会論文誌 Vol.48. No.8，pp.2736-2747(2007-08).
5) Tomohiro Nishida, Akira Harada, Tomoko Yoshida, Ryota Nakamura, Michio Nakanishi, Hirotoshi Toyoda, Kota Abe, Hayato Ishibashi, Toshio Matsuura："PEN: A Programming Environment for Novices --- Its Overview and Practical Lessons ---" ED-MEDIA(2008.06).
6) 有賀妙子，吉田智子:「ネットワークリテラシー教育のシラバスと教材研究」,『コンピュータと教育 50-4』情報処理学会，pp.25-32(1998.11).
7) 吉田智子:「本学における情報教育の現状と将来展望」,『教育のプリズム，ノートルダム教育』第 5 号，pp.73-119(2006.5).

Vol.2011-CE-109 No.7
2011/3/18

付録

付録 1： 2009 年度までの授業の練習問題一覧（第 3 回以外は 2010 年度もほぼ同様）

第 1 回　PEN の概要・逐次処理

練習問題 1.1：キーボードから 2 つの整数を入力し，その積および商を出力するプログラム（商の余りは切り捨て）
練習問題 1.2：ある鉄道の運賃は，乗車距離を a(km)にしたとき，(120+20×a)円である．乗車距離を入力して，運賃を求めるプログラム
練習問題 1.3：鉛筆をみんなで分けたい．鉛筆の数と人数を入力し，一人あたり何本もらえるか，また何本残るかのプログラム

第 2 回　条件分岐

練習問題 2.1：点数(x)を入力し，x が 60 以上なら「合格」と表示し，x が 60 未満なら「不合格」と表示するプログラム
練習問題 2.2：２つの値を受付け，大きい方を変数 max に，小さい方を変数を min に入れ，それらを表示するプログラム
練習問題 2.3：入力された数が 0 より小さい場合や 100 を超えた場合は「範囲外です」と表示し，60 以上なら「合格」，60 未満なら「不合格」と表示するプログラム

第 3 回　繰り返し処理

練習問題 3.1：1 から 100 までの整数の和を求めるプログラム
練習問題 3.2：2 つの整数 m と n を入力し，m から n までの整数の和を求めるプログラム（m は n より小さい整数とする）

第 4 回　図形描画

練習問題 4.1：例題 4.3 の半径 25 の赤い円描画プログラムの応用して，コンパスを使って描くような円のデザインを描画
練習問題 4.1：例題 4.4 のお花の描画を応用して，茎や葉っぱのある花を描画