解説

コンピュータを使って与えられた課題を解決するためには，その課題解決のために必要な情報処理の分析，そしてその情報処理をコンピュータに行わせるためのプログラムの作成が必要である。本問は，シフト暗号の解読という課題を例に，(i) 課題解決のために必要な情報と，それを得るために必要な処理の明確化，そして (ii) その処理を実現するプログラムの作成とその実行による課題解決，という一連の過程の中で**プログラミングの基本となる考え方や技法**を問う問題である。

シフト暗号は，ローマ皇帝シーザーも使っていたと言われている初歩的な暗号で，文字のシフト数がわかれば解読できる。問 1 では，暗号解読にはそのシフト数がわかればよい，という点を明確化させ，問 2 では（元の文が英文だと仮定したときに）シフト数は文字の出現頻度から推定できることから，各文字の出現頻度を計算するプログラムを作成し，その結果からシフト数を推定させ，問 3 で，その検証を行う復号プログラムを作成させている。問 2 の文字出現頻度の計算プログラムは，テキストデータマイニングの基本となる処理だが，一般にも多くの頻度分析プログラムに共通する計算手法である。

※高等学校の授業で多様なプログラミング言語が利用される可能性があることから，問題中で使用するプログラミング言語は，公平性を鑑みて，大学入試センター独自の日本語表記の疑似言語としている。

対応する情報 I の主な領域：(3)コンピュータとプログラミング　問題種：大問

学習指導要領（3）- 知・技 - イ
学習指導要領（3）- 思・判・表 - ウ
学習内容（3）- イ アルゴリズムとプログラム

第５問　次の文章を読み，後の問い(問 1～3)に答えよ。

シーザー暗号に代表される古典的な暗号化の方法であるシフト暗号はアルファベットの文字を決まった文字数分シフトさせて（ずらして）置き換える極めて単純な暗号手段である。T さんと M さんは授業で先生が出した課題であるシフト暗号で暗号化した暗号文をいかに解読するかを考えることにした。

問１　次の会話文を読み，空欄 アイ ～ キク に当てはまる数字をマークせよ。

課題　英文をシフト暗号で暗号化した以下の暗号文を解読しなさい。ただし，英文は全て小文字でアルファベット以外のスペースや数字，「'」「,」「.」「?」などは変換されていません。

（省略）‥‥nonsmkdo k zybdsyx yp drkd psovn, kc k psxkv bocdsxq zvkmo pyb dryco gry robo qkfo drosb vsfoc drkd dro xkdsyx wsqrd vsfo. sd sc kvdyqodrob psddsxq kxn zbyzob drkd go cryevn ny drsc. led, sx k vkbqob coxco, go mkx xyd nonsmkdo - go mkx xyd myxcombkdo - go mkx xyd rkvvyg - drsc qbyexn. dro lbkfo wox, vsfsxq kxn nokn, gry cdbeqqvon robo, rkfo myxcombkdon sd, pkb klyfo yeb zyyb zygob dy knn yb nodbkmd. dro gybvn gsvv vsddvo xydo, xyb vyxq bowowlob grkd go cki robo, led sd mkx xofob pybqod grkd droi nsn robo. sd ‥‥（省略）

図１　先生が出した課題

M さん：シフト暗号って，例えばアルファベットを５文字右にシフトした場合，文字「a」は文字「f」に，文字「x」はまず２文字シフトして右端に達した後一番左端に戻り３文字シフトした文字「c」に置き換わるやつだよね。暗号化された文字列の復号は，その逆，つまり左に５文字シフトすればできるね。

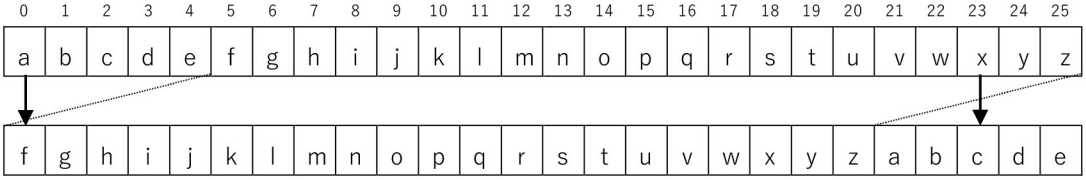

図２　５文字右シフトした場合の考え方

T さん:復号は必ずしも反対にシフトする必要はないよね。例えば 9 文字右にシフトされていた場合，復号するには 9 文字左にシフトでも良いけど，右に **アイ** 文字シフトすることもできるね。図 2 のようにアルファベットに 0～25 の番号を割り当てて考えてみると，暗号化して x 番目の文字になった時，復号は $x+$ アイ の値が **ウエ** 以下であれば $x+$ アイ 番の文字に置き換わるけど，ウエ より大きい場合は，$x+$ アイ $-$ **オカ** 番の文字に置き換えれば復号できるよね。

アイ ①⑦17

ウエ ②⑤25

オカ ②⑥26

M さん:暗号化で文字を何文字シフトしているか分かれば，この復号法で解読できるよね。どうやったら分かるかな。

T さん:すべての可能性，つまりシフトしない時を除いた **キク** 通りをプログラムで試せばいいんじゃない？

キク ②⑤25

M さん:この場合だと キク 通りで済むけども，大文字があったり，日本語のように文字種の数が多い言語ではとても効率が悪い方法だよ。英文であれば，単語に含まれる「a」とか「e」が多い気がするし，逆に「z」が含まれる単語は少ししか思いつかない。アルファベットの出現頻度を調べればある程度推測できるんじゃないかな。インターネットで調べてみようよ。

M さん:どうやら一般的な英文のアルファベットの出現頻度には図 3 のような傾向があるみたいだよ。

T さん:文字によって出現頻度に特徴があるね。暗号化された英文のアルファベットの出現頻度を調べれば，何文字シフトされているか推測することができそうだね。一つ一つ数え上げるのは大変だから数え上げるプログラムを考えてみるよ。

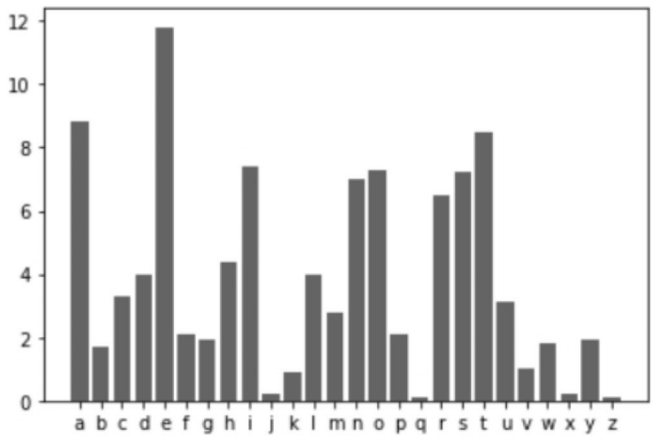

図3 出現頻度のグラフ（縦軸%）

学習指導要領（3）- 知・技 - イ
学習指導要領（3）- 思・判・表 - ウ
学習内容（3）- イ アルゴリズムとプログラム

問 2　次の会話文を読み，空欄 ケ ・ コ に当てはまる内容を，後の解答群のうちから一つずつ選べ。また，空欄 サシ に当てはまる数字をマークせよ。

Tさん：暗号化された英文のアルファベットの出現頻度を数え上げるプログラムを図5のように考えてみたよ。このプログラムでは，配列変数 Angoubun に暗号文を入れて，一文字ずつアルファベットの出現頻度を数え上げて，その結果を配列変数 Hindo に入れているんだ。Hindo[0]が「a」，Hindo[25]が「z」に対応しているよ。

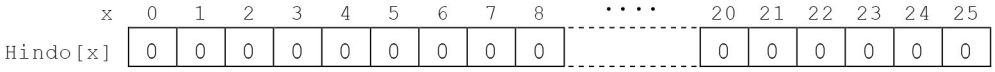

図4　アルファベットの出現頻度を数え上げる配列

```
(01)Angoubun = ["p","y","e","b",…(省略)…"k","b","d","r","."]
(02)配列 Hindo のすべての要素に 0 を代入する
(03)i を 0 から 要素数(Angoubun)-1 まで 1 ずつ増やしながら:
(04)  bangou = 差分( ケ )          ケ ⓪Angoubun[i]
(05)  もし bangou != −1 ならば:
(06)    コ = コ + 1                コ ④Hindo[bangou]
(07)表示する(Hindo)
```

図5　出現頻度を求めるプログラム

【関数の説明】

要素数(値)…配列の要素数を返す。
例：Data=["M","i","s","s","i","s","s","i","p","p","i"]の時
要素数(Data)は 11 を返す

差分(値)…アルファベットの「a」との位置の差分を返す
値がアルファベット以外の文字であれば－1を返す
例：**差分**("e")は 4 を，**差分**("x")は 23 を返す
差分("5")や**差分**(",")は－1を返す

M さん:これでアルファベットの出現頻度が調べられるね。それで結果はどうなったの？

T さん:このプログラムで得られた配列 Hindo をグラフ化してみたよ（図６）。

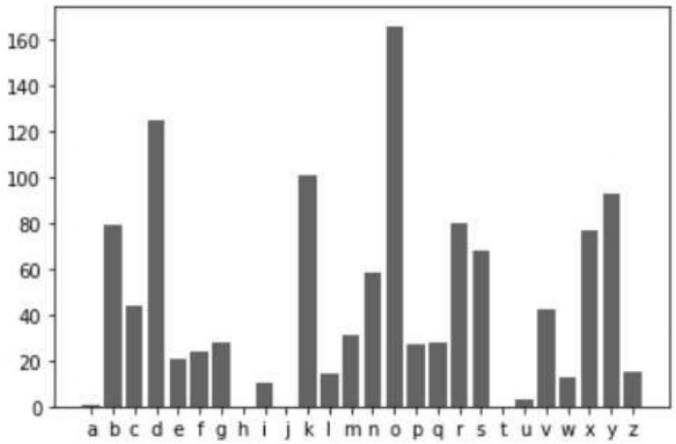

図６　アルファベットと配列 Hindo のグラフ表示

M さん:このアルファベットの出現頻度を見ると，「o」「d」「k」「y」が多いね。逆に出現頻度がない「a」「h」「j」「t」も手掛かりになるね。図３と照らし合わせると，この暗号化された文字列は右に［サシ］文字シフトしていると考えられるね。

サシ　①⓪10

T さん:うん，でもそれが正しいか，実際にプログラムを作って復号してみようよ。

［ケ］・［コ］の解答群

⓪ Angoubun[i]　① Angoubun[i-1]　② Angoubun[bangou]
③ Angoubun[bangou-1]　④ Hindo[bangou]　⑤ Hindo[bangou-1]
⑥ Hindo[i]　⑦ Hindo[i-1]

学習指導要領（3）- 知・技 - イ
学習指導要領（3）- 思・判・表 - ウ
学習内容（3）- イ アルゴリズムとプログラム

問 3　次の会話文の空欄［ス］～［チ］に当てはまる内容を，後の解答群のうちから一つずつ選べ。

T さん：暗号文を一文字ずつ復号して表示するプログラムができたよ（図 7）。

M さん：なるほど，復号も右にシフトで考えているんだね。実行してみたら読み取れる英文になったの？

```
(01)Angoubun = ["p","y","e","b",…(省略)…"k","b","d","r","."]
(02)配列変数 Hirabun を初期化する
(03)hukugousuu = 26 - [サシ]
(04)i を 0 から 要素数(Angoubun)-1 まで 1 ずつ増やしながら:
(05)  bangou = 差分([ケ])
(06)  もし bangou != -1 ならば:
(07)    もし [ス] <= 25 ならば:
(08)      Hirabun[i] = 文字([ス])
(09)    そうでなければ:
(10)      Hirabun[i] = 文字([セ])
(11)  そうでなければ:
(12)    Hirabun[i] = [ソ]
(13)表示する(Hirabun)
```

ス ⓪bangou+hukugousuu
※
セ ③bangou+hukugousuu-26
ソ ⑥Angoubun[i]

図7　暗号文を復号するプログラム

【関数の説明】

文字(値)…番号の値に対するアルファベットの文字を返す。
　　値が 0 以上 25 以下でなければ「アルファベットでない」を返す
例：**文字**(4)は「e」を，**文字**(23)は「x」を返す
　　文字(-1)や**文字**(27)は「アルファベットでない」を返す

T さん：うん，復号したらこんな英文が表示されたよ。正しい英単語に変換されているみたいだから推測は当たっていたね。

four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.now we are engaged in a great civil war, testing whether that nation, or any nation ·····（省略） ····· last full measure of devotion - that we here highly resolve that these dead shall not have died in vain - that this nation, under god, shall have a new birth of freedom - and that government of the people, by the people, for the people, shall not perish from the earth.

M さん:これって有名なリンカーンのゲティスバーグ演説じゃない。ほら最後のところ有名なフレーズだよね。

T さん:先生,課題ができました。元の英文はリンカーンのゲティスバーグ演説ですね。プログラムで文字の出現頻度を調べて,シフトされた文字数を推測しました。復号はこのプログラムで変換してみました。

先生:よくできたね,素晴らしい!このプログラムはもっと簡単にできるね。この(07)～(10)の※部分は工夫すれば1行にまとめられるよ。ヒントは余りを求める算術演算子%を使うんだ。

T さん:えっ, 1行ですか?・・・・・分かった!

```
Hirabun[i] = 文字(［タ］%［チ］)
```

とすればもっと簡潔にできたんだ。

タ ①(bangou+hukugousuu)
チ ①26

先生:素晴らしい!

［ス］～［ソ］の解答群

⓪ bangou+hukugousuu　① bangou
② hukugousuu　③ bangou+hukugousuu-26
④ bangou+hukugousuu-25　⑤ hukugousuu-26
⑥ Angoubun[i]　⑥ Hirabun[i]
⑧ Angoubun[i+hukugousuu]

［タ］の解答群

⓪ bangou+hukugousuu　① (bangou+hukugousuu)
② i+hukugousuu　③ (i+hukugousuu)
④ hukugousuu+26　⑤ (hukugousuu+26)

［チ］の解答群

⓪ 25　① 26　② bangou　③ hukugousuu