文部科学省後援

情報システム試験

平成28年度前期 情報検定

＜実施　平成28年9月11日（日）＞

プログラミングスキル

（説明時間　10：00～10：10）

（試験時間　10：10～11：40）

・試験問題は試験開始の合図があるまで開かないでください。

・解答用紙（マークシート）への必要事項の記入は，試験開始の合図と同時に行いますので，それまで伏せておいてください。
・試験開始の合図の後，次のページを開いてください。**＜受験上の注意＞**が記載されています。必ず目を通してから解答を始めてください。
・試験問題は，すべてマークシート方式です。正解と思われるものを１つ選び，解答欄の()をＨＢの黒鉛筆でぬりつぶしてください。２つ以上ぬりつぶすと，不正解になります。
・辞書，参考書類の使用および筆記用具の貸し借りは一切禁止です。
・電卓の使用が認められます。ただし，下記の機種については使用が認められません。

＜使用を認めない電卓＞

１．電池式（太陽電池を含む）以外の電卓
２．文字表示領域が複数行ある電卓（計算状態表示の一行は含まない）
３．プログラムを組み込む機能がある電卓
４．電卓が主たる機能ではないもの
　＊パソコン（電子メール専用機等を含む），携帯電話（ＰＨＳ），スマートフォン，タブレット，電子手帳，電子メモ，電子辞書，翻訳機能付き電卓，音声応答のある電卓，電卓付き腕時計，時計型ウェアラブル端末等
５．その他試験監督者が不適切と認めるもの

一般財団法人　職業教育・キャリア教育財団

＜受験上の注意＞

１．この試験問題は29ページあります。ページ数を確認してください。
乱丁等がある場合は，手をあげて試験監督者に合図してください。
※問題を読みやすくするために空白ページを設けている場合があります。

２．解答用紙（マークシート）に，受験者氏名・受験番号を記入し，受験番号下欄の数字をぬりつぶしてください。正しく記入されていない場合は，採点されませんので十分注意してください。

３．試験問題についての質問には，一切答えられません。自分で判断して解答してください。

４．試験中の筆記用具の貸し借りは一切禁止します。筆記用具が破損等により使用不能となった場合は，手をあげて試験監督者に合図してください。

５．試験を開始してから30分以内は途中退出できません。30分経過後退出する場合は，もう一度，受験番号・マーク・氏名が記載されているか確認して退出してください。なお，試験終了５分前の合図以降は退出できません。試験問題は各自お持ち帰りください。

６．試験後にお知らせする合否結果（合否通知），および合格者に交付する「合格証・認定証」はすべて，Ｗｅｂページ（ＰＣ，モバイル）での認証によるデジタル「合否通知」，デジタル「合格証・認定証」に移行しました。
①団体宛にはこれまでと同様に合否結果一覧ほか，試験結果資料一式を送付します。
②合否等の結果についての電話・手紙等でのお問い合わせには，一切応じられませんので，ご了承ください。

＜問題の構成＞

必須問題　全員解答

問題１～問題４	2 ページ～14 ページ

選択問題　次の問題から１問選択し解答せよ。

（選択した問題は解答用紙「選択欄」に必ずマークすること）

※選択欄にマークがなく，解答のみマークした場合は採点を行いません。

・Ｃ言語の問題	16 ページ～20 ページ
・表計算の問題	21 ページ～26 ページ
・アセンブラの問題	27 ページ～29 ページ

必須問題

問題１　次のデータ構造に関する説明を読み，各設問に答えよ。

スタック構造では，データをスタックに入れる PUSH 操作とスタックからデータを取り出す POP 操作がある。PUSH 操作,POP 操作ではスタックの最上段の位置を示すスタックポインタ(SP)を使用する。SP の初期値は－１とし，スタックにデータが入っているときの SP は 0 以上となる。

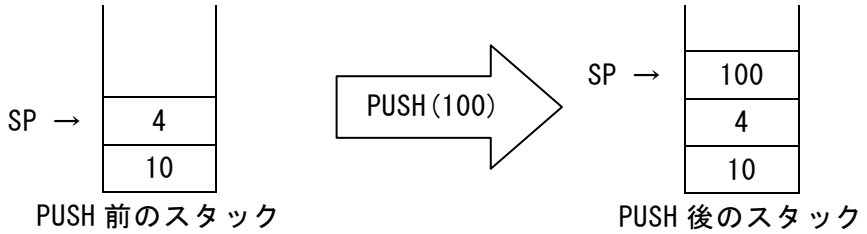


図１　PUSH の動作

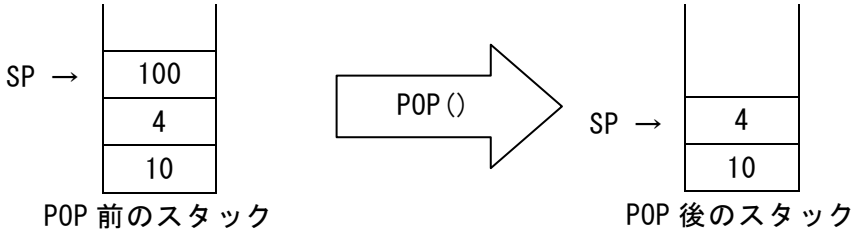


図２　POP の動作

キュー構造では，データをキューに入れる INQ 操作とキューからデータを取り出す DEQ 操作がある。

＜設問１＞　次のスタック構造に関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

スタック領域を「STACK」，スタックポインタを「SP」としたとき，PUSH 操作と POP 操作は次のようになる。なお，領域内を参照する場合は STACK[SP] と表す。

・PUSH(ａ)　…　データ a をスタックに入れる。

［(1)］
STACK[SP]　←　a

・POP() …　スタックの最上段のデータをｂに取り出す。

ｂ　←　STACK[SP]
(2)

このように，スタック構造でのデータの出し入れは(3)となる。

(1) ～ (3) の解答群

ア．FIFO　　イ．LIFO
ウ．SP ← SP + 1　　エ．SP ← SP - 1
オ．STACK[SP] ← STACK[SP] + 1　　カ．STACK[SP] ← STACK[SP] - 1

＜設問２＞　次のスタック構造とキュー構造に関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

キュー構造で，データａをキューに入れる操作を INQ(a)，キューからデータを取り出す操作を DEQ()とする。スタックの操作は設問 1 と同様とした場合，次の①～⑧のように操作をすると，X には(4)が入り，Y には(5)が入る。

［データ操作］
① INQ(P)
② INQ(Q)
③ INQ(R)
④ INQ(S)
⑤ PUSH(DEQ())
⑥ PUSH(DEQ())
⑦ X ← POP()
⑧ Y ← DEQ()

(4)，(5) の解答群

ア．P　　イ．Q　　ウ．R　　エ．S

問題２　次の整列アルゴリズムに関する記述を読み，各設問に答えよ。

［バケットソートの説明］

1 次元配列 A[0]～A[14]に 10 進数 1 桁の数値データが 15 個格納されている(図１)。なお，同じデータは最大でも 5 個とする。

このデータを，バケットソート法により昇順に整列する。バケットソート法とは，データを整列するためのアルゴリズムであり，次の手順１，手順２により整列する。

手順１：配列 dt から順にデータを取り出し，配列 bkt に格納する。例えばデータが 5 なら bkt[5]の行へ bkt[5][0]， bkt[5][1]，…の順に格納する。このとき配列 cnt に，同じ値のデータの出現回数をカウントする(図２)。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
dt	5	1	8	4	5	2	0	1	7	8	0	4	5	9	5

図１　配列 dt の内容

		0	1	2	3	4
bkt	0	0	0			
	1	1	1			
	2	2				
	3					
	4	4	4			
	5	5	5	5	5	
	6					
	7	7				
	8	8	8			
	9	9				

	0	1	2	3	4	5	6	7	8	9
cnt	2	2	1	0	2	4	0	1	2	1

図２　手順１実行後の各配列の内容

手順２：配列 bkt の 0 行から順にデータを取り出し，配列 dt の先頭から順に格納する(図３)。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
dt	0	0	1	1	2	4	4	5	5	5	5	7	8	8	9

図３　手順２実行後の配列 dt の内容

＜設問１＞　次のバケットソートに関する流れ図中の［　　］に入れるべき適切な字句を解答群から選べ。

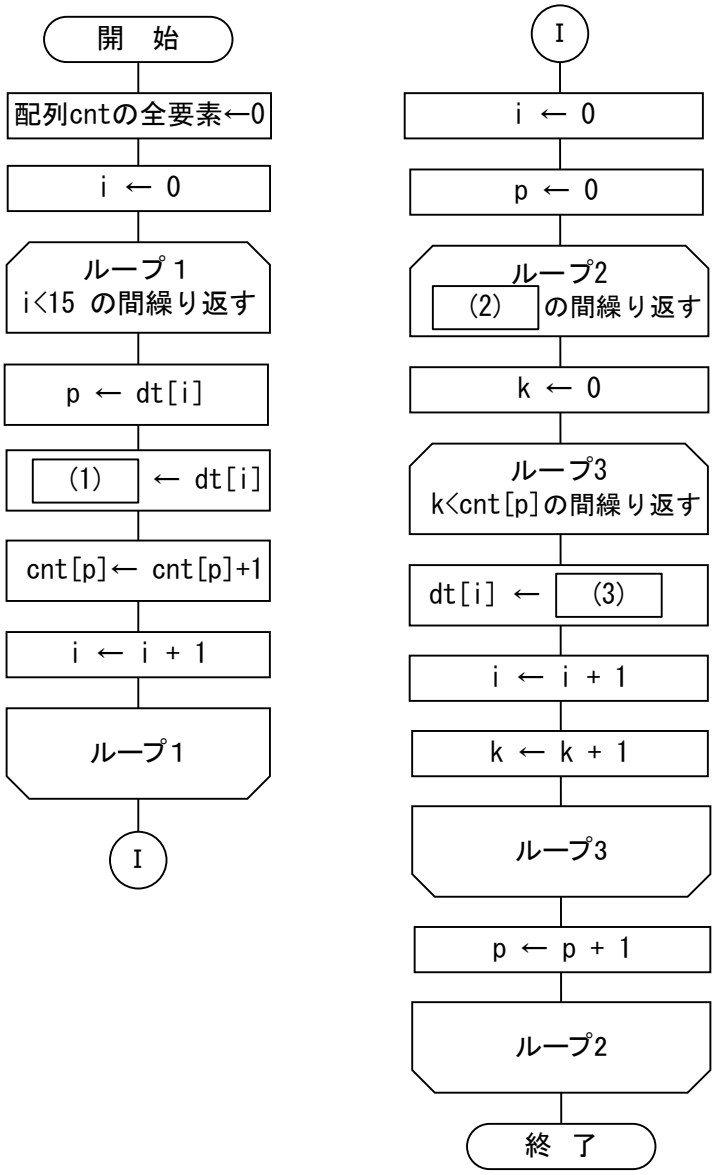


図４　バケットソートの流れ図

(1)，(3) の解答群

ア．bkt[i][p]　　イ．bkt[p][k]

ウ．bkt[i][cnt[i]]　　エ．bkt[p][cnt[p]]

(2) の解答群

ア．i ＜ 10　　イ．i ＜ 15　　ウ．p ＜ 10　　エ．p ＜ 15

＜設問２＞　次のバケットソートの問題点に関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

図４のバケットソートは，バケット（配列 bkt）の大きさがデータの範囲や個数に大きく左右され，バケットの未使用領域も増加するという問題点がある。

例えば，データを 10 進数 2 桁以内とし，データ数が 50 個の場合，図４の流れ図の方式では配列 bkt は，［(4)］行必要であり，また列数も，同じ値のデータの個数が未知の場合［(5)］列必要である。

(4)，(5) の解答群

ア．10　　イ．50　　ウ．99　　エ．100

問題３　文字列の置換に関する次の記述を読み，各設問に答えよ。

ある文字列の中から置換される文字列を検索し，その文字列を別の文字列（置換する文字列）に変更する処理を考える。例えば，最初の文字列が"ABCDEFG"，置換される文字列が"CD"，置換する文字列を"XXX"とすれば，置換後の文字列は"ABXXXEFG"となる。

［置換処理の条件］

・置換前の文字列をin，置換される文字列をbefore，置換する文字列をafter，置換後の文字列をoutとし，inからoutへ１文字ずつコピーしながら置換を行う。

例：

in	ABCDEFG
before	CD
after	XXX

→ out ABXXXEFG

・in，before，after，outは一次元配列であり，１つの要素に１文字ずつ格納される。なお，outは置換が完了した文字列を格納するために十分な領域が割当てられているものとし，配列の要素位置は0から始まる。

・inの文字数をi_len，beforeの文字数をb_len，afterの文字数をa_len，outの文字数をo_lenとする。

＜設問１＞　次の文字列の検索に関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

配列inに格納された文字列から配列beforeに格納された文字列を検索することを考える。配列inと配列beforeの文字が連続して一致するかを調べる。図１のように，配列beforeの内容がすべて一致すれば文字列が検索できたことになる。

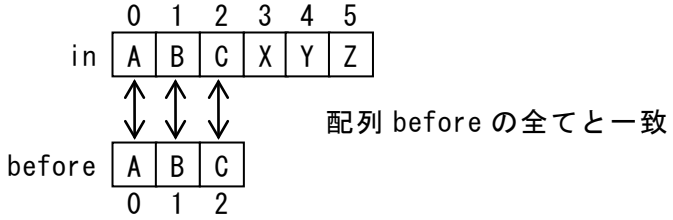


図１　置換される文字列が一致する場合

また，図２のように，途中で一致しない場合もある。

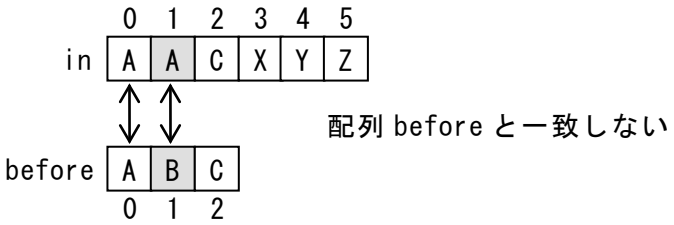


図２　置換される文字列が一致しない場合の例１

図２の場合，in[0]とbefore[0]の比較から始めたが，in[1]とbefore[1]で一致しなかったため，次のin[2]とbefore[2]の比較を中止し，in[1]とbefore[0]から検索をやり直す。

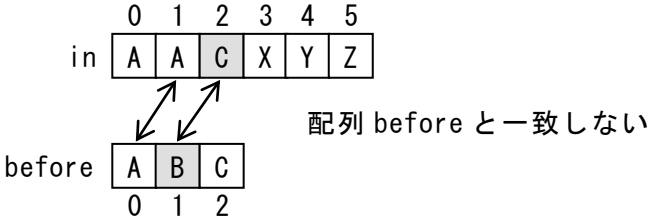


図３　置換される文字列が一致しない場合の例２

図３はin[1]とbefore[0]から検索をやり直したものであるが，途中で一致しない文字が出現するので検索を中断し，in[2]とbefore[0]から検索をやり直す。このように，途中で一致しない文字が出現した場合，検索を開始する配列inの位置をずらしながら処理する。

ここで，配列inの検索開始位置をPとし，in[P+K]とbefore[K]（K=0，1，2，…）の比較を行うとする。途中で不一致となった場合はin[　(1)　]とbefore[0]からやり直す。なお，比較をする時のKは，最大で　(2)　になる。また，in[P+K]とする要素位置が配列inの大きさ（文字数）を超えてはならない。そのため，P+Kが　(3)　を超えないように制御しなければならない。

(1) の解答群

ア．P ＋ 1　　イ．P ＋ 2　　ウ．P ＋ 3　　エ．P ＋ 4

(2)，(3) の解答群

ア．b_len － i_len　　イ．b_len － 1　　ウ．i_len － b_len　　エ．i_len － 1

＜設問２＞　次の文字列の置換に関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

配列beforeの文字列が，配列inの中に見つかったかどうかにより，配列outへコピーする文字が異なる。

配列beforeと全て一致する部分が配列inに見つかった場合は，配列afterを全て配列outへコピーする。

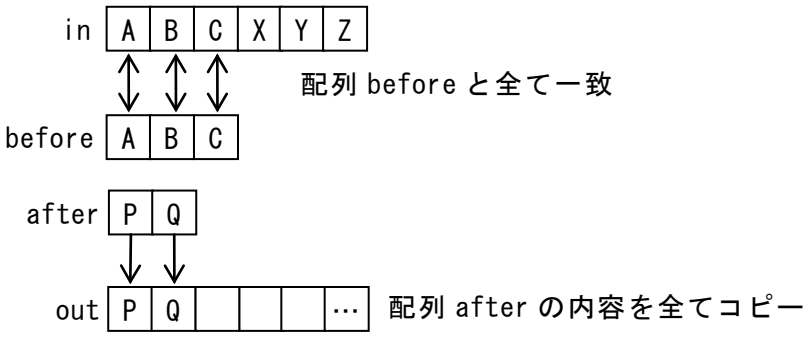


図４　置換される文字列と一致した場合

検索の途中で一致しない文字が出現した場合は，配列inの探索開始位置にある文字をコピーする。

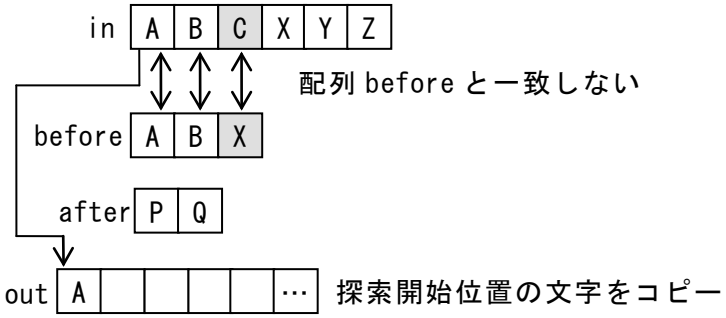


図５　置換される文字列と一致しなかった場合

これらの操作後に検索をやり直す場合，配列inの検索開始位置も変わってくる。

今，in[P]とbefore[0]の比較から検索を行ったとする。次の検索は，配列beforeの文字列が一致すればin[　(4)　]とbefore[0]との比較，そうでなければin[　(5)　]とbefore[0]の比較からとなる。

(4)，(5)の解答群

ア．P + 1　　イ．P + b_len

ウ．P + a_len　　エ．P + i_len - a_len

＜設問３＞　次の文字列の置換を行う流れ図中の［　　］に入れるべき適切な字句を解答群から選べ。

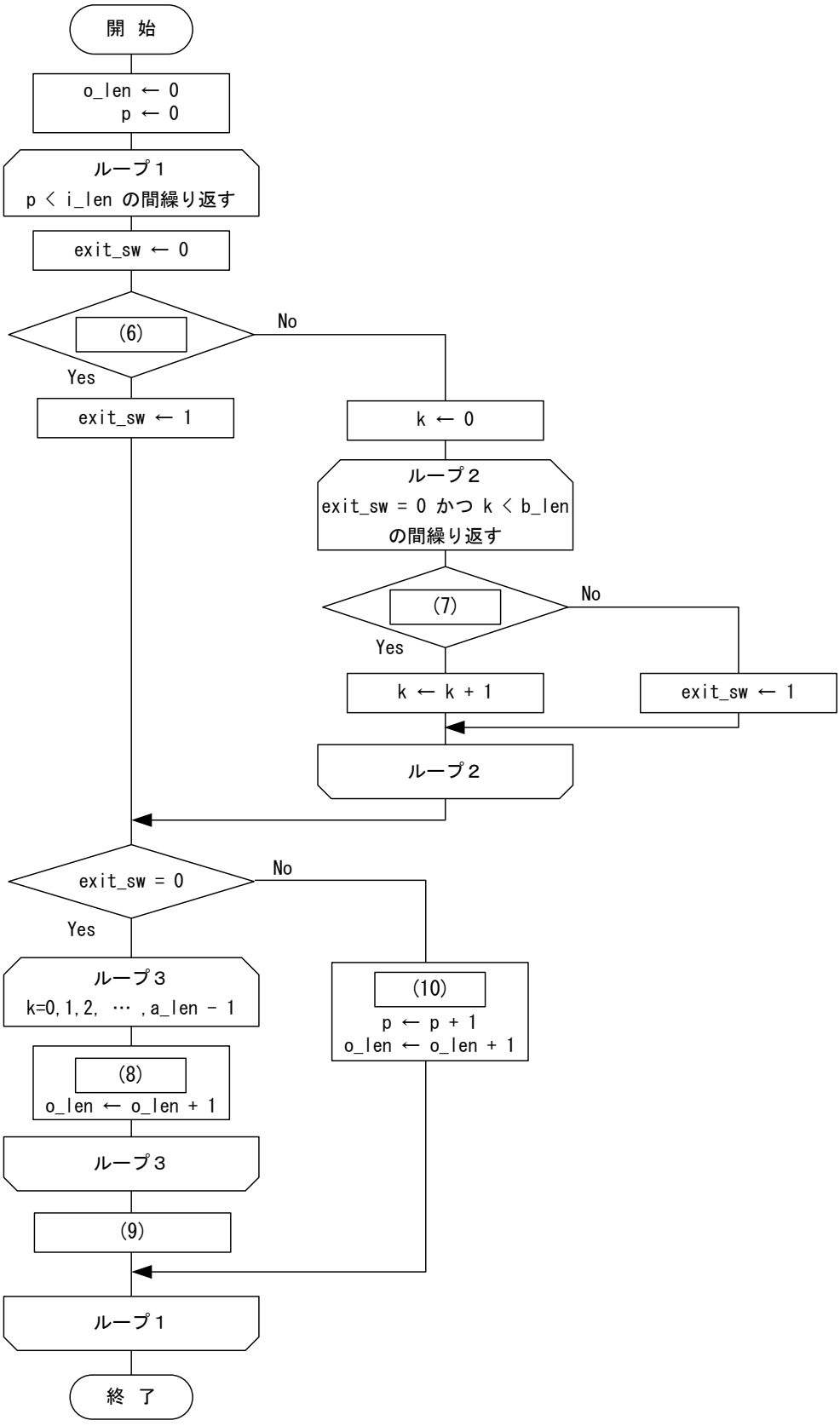


図６　置換処理の流れ図

(6) の解答群

ア. p > b_len - i_len　　イ. p > b_len

ウ. p > i_len - b_len　　エ. p > i_len

(7) の解答群

ア. in[k] = before[p]　　イ. in[k] = before[p + k]

ウ. in[p + k] = before[k]　　エ. in[p + k] = before[p]

(8) の解答群

ア. out[o_len] ← after[k]　　イ. out[o_len] ← after[p]

ウ. out[o_len] ← after[p - k]　　エ. out[o_len] ← after[p + k]

(9) の解答群

ア. p ← p + b_len　　イ. p ← p + i_len

ウ. p ← p + o_len　　エ. p ← p + a_len

(10) の解答群

ア. out[o_len] ← in[k]　　イ. out[o_len] ← in[k - p]

ウ. out[o_len] ← in[k + p]　　エ. out[o_len] ← in[p]

問題４　次のプログラムの説明および擬似言語の記述形式の説明を読み，設問に答えよ。

［プログラムの説明］

要素数が N 個の１次元配列 DAT[k]（k=0，1，…，N-1）から二分探索法によりデータを探索する関数 B_search である。なお，見つかった場合はその位置（添字の値）を，見つからなかった場合は-1 を，変数 P に設定するものとする。

二分探索法とは，昇順，または降順に整列済みである配列を利用した探索方法で，探索しようとする値と１次元配列の中央の値を比べ，その大小関係によって探索範囲を狭くして目的のデータを検索するものである。

ここで，配列の大きさは N に，探したいデータは X に，データは１次元配列 DAT に昇順に格納済みとし，二分探索法の手順をⅠ～Ⅲに示す。

Ⅰ．探索範囲の先頭要素の添字を Low，末尾要素の添字を High とする。なお，初期値は，Low＝0，High＝N-1 である。

Ⅱ．探索範囲の中央要素となる DAT[M]と比較する。ただし，M＝(Low+High)÷2 とし，小数点以下は切り捨てる。

DAT[M]＜X なら，Low＝M+1 とし，次の探索範囲を，配列の要素位置が M より大きい方とする。

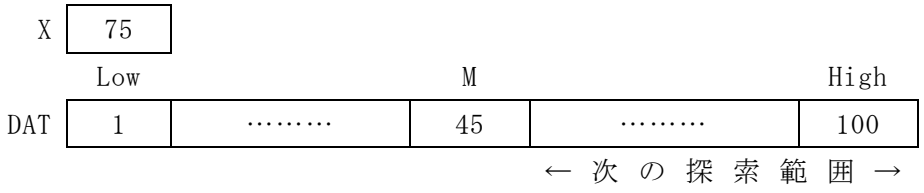


図１　比較例１

DAT[M]＞X なら，High＝M-1 とし，次の探索範囲を，配列の要素位置が M より小さい方とする。

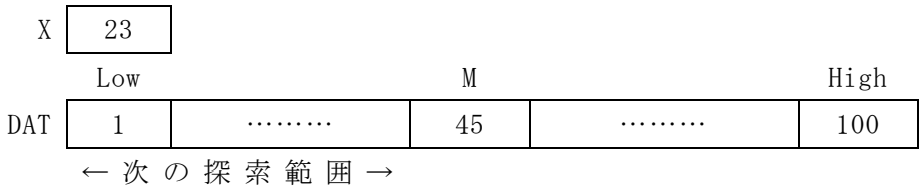
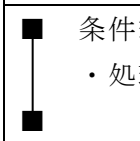


図２　比較例２

DAT[M]＝X なら，見つかった時の処理をして，探索を終了する。

Ⅲ．Low＞High となるまで，Ⅱを繰り返す。Low＞High の場合は，探したいデータ X と同じ値が配列 DAT に存在しないことになる。

［擬似言語の記述形式の説明］

記述形式	説明
○	手続き，変数などの名前，型などを宣言する
・変数 ← 式	変数に式の値を代入する
/* 文 */	注釈を記述する
	選択処理を示す。 条件式が真の時は処理１を実行し， 偽の時は処理２を実行する。
	前判定繰り返し処理を示す。 条件式が真の間，処理を実行する。

［プログラム］

○**B_search**（整数型：**N**，文字型：**DAT[]**，整数型：**X**）

○整数型：**Low, High, m, P**

```
・Low ← 0
・High ← N - 1
・P ← -1
/* 配列の中から X を探索する */
■ P = -1 かつ (1)
  ・M ← (Low + High)/2
  ▲ DAT[M] = X                α
    ・P ← M
    ・break          /* 繰り返し処理から抜け出す */
    ▲ DAT[M] < X
      ・(2)
    ▼
      ・(3)
  ▼
■
```

＜設問１＞　プログラム中の［　　］に入れるべき適切な字句を解答群から選べ。

(1) の解答群

ア．**Low ＜ High**　　イ．**Low ≦ High**

ウ．**Low ＞ High**　　エ．**Low ≧ High**

(2)，(3) の解答群

ア．**High ← M - 1**　　イ．**High ← M + 1**

ウ．**Low ← M - 1**　　エ．**Low ← M + 1**

＜設問２＞　配列 DAT の内容が次のような場合，プログラム中のαを実行するときの変数 Low，High，M をトレースした表の［　　］に入れるべき適切な字句を解答群から選べ。

X | 21 |

	0	1	2	3	4	5	6	7	8	9
DAT	1	2	3	5	8	13	21	34	55	89

表　トレースの内容

順番	Low	High	M
1	0	9	4
2	(4)		
3	5	6	5
4	(5)		

(4)，(5) の解答群

ア．	5	6	5
イ．	5	9	7
ウ．	6	6	6
エ．	6	8	7

＜ 選 択 問 題 ＞

選択問題は問題から１つ選択し解答せよ。
選択した問題は必ず，解答用紙「選択欄」にマークすること。
※選択欄にマークがなく，解答のみの場合は採点を行いません。

各構成は以下のとおり。

選択問題

選択問題　C言語の問題

次のＣ言語プログラムの説明を読み，各設問に答えよ。

［プログラムの説明］

都市間の距離をもとにした最短経路を求める副プログラムroute_searchである。

都市間の経路が図１のような場合，route_searchに渡される経路情報は図２のようになり，都市番号0から3までの最短経路を計算する。

なお，都市の数がnの場合，最初に出発する都市番号は0であり，最後に到着する都市番号はn-1とする。また，最短経路を通った時の距離は9999より小さいものとする。

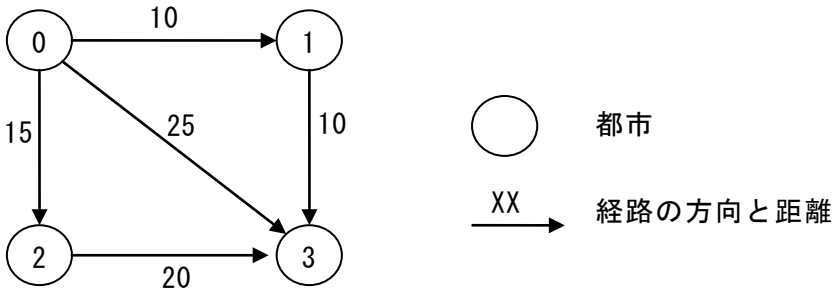


図１　４つの都市を結ぶ経路

出発都市番号	到着都市番号	距離
0	1	10
0	2	15
0	3	25
1	3	10
2	3	20

図２　route_searchに渡される５つの経路情報

［最短経路を求める手順］

経路の数をm,都市の数をnとし,最短経路を求めるために利用する配列をdistanceとpointとする。distanceは都市番号0から各都市番号までの最短距離を，pointはdistanceが変更されたときの都市番号を格納する。それぞれ配列の要素数はnである。

１．distance[0]を0，distance[0]以外を9999， pointの全要素を-1で初期化する。

２．配列distanceの変更が発生する間，以下の処理を繰り返す。

m本の経路すべてについて,「"出発都市番号"の最短距離と"距離"を加えた値」,「"到着都市番号"の最短距離」の２つを比較する。前者が小さい場合に，配列distanceの変更（"到着都市番号"の位置に前者の値を設定）と，配列pointの変更（"到着都市番号"の位置に"出発都市番号"を設定）をする。

３．配列pointに求められた都市番号を反対にたどることで最短経路がわかる。

＜設問１＞　次の最短経路を求める処理のトレースに関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

図１および図２の内容から最短経路を求める手順を検証する。

都市の数が4なので，配列distanceとpointは4つの要素を持つ配列とし，初期値をそれぞれ設定すると，図３のようになる（[最短経路を求める手順］の１）。

	0	1	2	3
distance	0	9999	9999	9999
point	-1	-1	-1	-1

図３　最短経路を求める処理で使用する配列の初期設定

図２の経路情報から１件目の情報を取り出してみる。

出発都市番号	到着都市番号	距離
0	1	10

最短経路を求める手順の２に記述されている「"出発都市番号"の最短距離と"距離"を加えた値」は，出発都市番号が0であるため，distance[0] + 10 = 10　となる。

また,「"到着都市番号"の最短距離」は,到着都市番号が1であるため,distance[1]に格納されている9999である。

この場合，前者が小さいので，distanceとpointoの変更を行う。到着都市番号の位置は1，小さい値（前者の値）は10，出発都市番号は0であるから，distance[1]を10，point[1]を0に変更する。

	0	1	2	3
distance	0	10	9999	9999
point	-1	0	-1	-1

図４　１件目の経路情報で変更した内容

同様に，２件目の経路データについても考える。

図２の経路情報の２件目は次のようになっている。

出発都市番号	到着都市番号	距離
0	2	15

「"出発都市番号"の最短距離と"距離"を加えた値」は，distance[0] + 15 = 15であり，「"到着都市番号"の最短距離」はdistance[2]に格納されている9999である。この場合も前者が小さいので，distance[2]を15，point[2]を0に変更する。

	0	1	2	3
distance	0	10	15	9999
point	-1	0	0	-1

図５　２件目までの経路情報で変更した内容

以下，３件目以降の経路データから比較する対象は次のようになる。

３件目：distance[0]に25を加えた値とdistance[3]を比較

４件目：distance[1]に10を加えた値とdistance[3]を比較

５件目：distance[2]に20を加えた値とdistance[3]を比較

この結果，配列distanceとpointは，次の図のようになる。

	0	1	2	3
distance	0	(1)	(2)	(3)
point	-1	0	0	1

図６　1回目の処理で更新した結果

なお，配列pointの最後の要素位置からリスト構造のポインタのようにたどることで最短経路がわかる。

配列pointが図６のような場合，最後に到着する都市番号が3なのでpoint[3]からたどり始め，point[3]の値が1，point[1]の値が0と，最初に出発する都市番号0までたどり，反対に並べた0→1→3が最短経路となる。

(1) ～ (3) の解答群

ア．10　　イ．15　　ウ．20　　エ．25　　オ．30

[route_search 関数の説明]

引　数：m（経路数），n（都市数），route（経路情報），ret（最短経路）

機　能：都市番号 0 から都市番号 n-1 までの経路の中で最短距離の経路を求め，ret に都市番号 0 から順番に経由する都市番号を格納する。ret の最後は -1 を格納する。

戻り値：なし

＜設問２＞　次のプログラム中の［　　］に入れるべき適切な字句を解答群から選べ。

［プログラム］

```
struct ROUTE {
    int    from;      /* 出発都市番号 */
    int    to;        /* 到着都市番号 */
    int    cost;      /* 距離 */
};

void  route_search(int m, int n, struct ROUTE *route, int *ret) {
    int i, k, from, to, chk_value, loop_sw, update_sw, wk;
    int *point, *distance;
    /* 配列distanceとpointの領域を確保 */
    distance = (int *)malloc(sizeof(int) * n);
    point = (int *)malloc(sizeof(int) * n);
    /* distanceとpointの初期化 */
    for(i = 0; i < n; i++) {
        distance[i] = 9999;
        point[i] = -1;
    }
    distance[0] = 0;
    loop_sw = 0;
    while(loop_sw == 0) {
        update_sw = 0;
        for(i = 0; i < m; i++) {
            from = route[i].from;
            to   = route[i].to;
            chk_value = [ (4) ] + route[i].cost;
            if (chk_value < [ (5) ]) {
                [ (5) ] = chk_value;
                [ (6) ];
                update_sw = 1;
            }
        }
        if ( [ (7) ] ) {
            loop_sw = 1;
        }
    }
```

```
    /* pointを末尾からたどって最短経路をretへ格納（逆並び） */
    ret[0] = n - 1;          /* 最後の到着都市番号を格納 */
    i = n - 1;               /* ポインタの設定 */
    k = 1;                   /* retの添え字初期化 */
    while(i != 0) {
         ret[k] = point[i];
         [ (8) ];
         k++;
     }
     /* retの並びを反対にする */
     for(i = 0; i < k/2; i++) {
         wk = ret[i];
         ret[i] = ret[k - i - 1];
         ret[k - i - 1] = wk;
     }
     ret[k] = -1;
}
```

(4)，(5) の解答群

ア. distance[from]　　イ. distance[m-1]

ウ. distance[n-1]　　エ. distance[to]

(6) の解答群

ア. point[from] = from　　イ. point[from] = to

ウ. point[to] = from　　エ. point[to] = to

(7) の解答群

ア. loop_sw == 0　　イ. loop_sw == 1

ウ. update_sw == 0　　エ. update_sw == 1

(8) の解答群

ア. i = point[i]　　イ. i = point[k]

ウ. i++　　エ. i--

選択問題　表計算の問題

この問題で使用する表計算ソフトの仕様は下記のとおりである。

IF 関数

条件が真のときに真の場合，偽のときに偽の場合の計算結果や値を返す。

書式：IF(条件，真の場合，偽の場合)

MATCH 関数

検索範囲から検索値が存在するセルの相対的な位置を返す。位置は，1 から始まる相対的な値である。検索範囲は 1 行または 1 列であり，A1:B10 のように複数行または複数列での指定はできない。検索の型は 0 を指定すると検索値と完全に一致する値を検索する。

書式：MATCH(検索値，検索範囲，検索の型)

MAX 関数

範囲の中に含まれる数値の最大値を返す。

書式：MAX(範囲)

MIN 関数

範囲の中に含まれる数値の最小値を返す。

書式：MIN(範囲)

ROUNDDOWN 関数

指定した桁で値を切り捨てる。桁数が正の数であれば小数点以下，負の数であれば小数点以上の桁になる。例えば，1 にすると小数点以下第 2 位以下の桁を切り捨てて小数点以下第 1 位までを表示し，-1 にすると 1 の位以下の桁を切り捨てる。

書式：ROUNDDOWN(式または値，桁数)

VLOOKUP 関数

検索値を左端に含む行を範囲の中から検索し，指定した列位置の値を返す。検索の型に 0 を指定すると検索値と完全に一致する値を検索し，1 を指定すると検索値と一致する値がない場合に，検索値未満で一番大きい値を検索する。

書式：VLOOKUP(検索値，範囲，列位置，検索の型)

式

＝に続いて計算式や関数などを入力する。

セル番地の絶対参照

セル番地に＄を付けることで，絶対番地（絶対参照）を表す。

別シートの参照

ワークシート名に「！」を付けてセル位置を指定することにより別シートを参照できる。
例：シート名「集計」のセル A1 を参照する場合は，「集計!A1」と記述する。

Ｊ社では給与体系を刷新することになり，担当のＡ君は社員の給与に関する情報を表計算ソフトでまとめることにした。なお，社員は20名いる。

Ｊ社では，基本給以外に給与に含む手当てがいくつかあり，次のようになっている。

［役職手当］

次の表１の金額が役職を持つ社員に支給される。

表１　役職手当

役職名	金額
部長	5万円
次長	4万円
課長	3万円
係長	2万円
主任	1万円

［家族手当］

配偶者がいる場合は1万5千円を支給する。また，扶養している18歳までの子どもがいる場合，2人目までは1人につき1万円，3人目以降は1人につき5千円を支給する。

［通勤手当］

公共交通機関を利用して通勤する場合は１カ月の定期代を支給する。

自家用車（オートバイを含む）または自転車で通勤する場合は，次の表２のように距離に応じて支給する。

表２　自家用車または自転車通勤の場合の手当

距離	自家用車	自転車
２km未満	支給しない	支給しない
２km以上　10km未満	4,000円	2,000円
10km以上　15km未満	7,000円	4,000円
15km以上　25km未満	12,000円	7,000円
25km以上　35km未満	18,000円	10,000円
35km以上	24,000円	14,000円

複数の交通手段で通勤する場合は，それらを全て合算した金額を支給する。

ただし，いずれの場合も10万円を上限とする。

［住宅手当］

賃貸住宅に居住している場合は3万円を上限として家賃の半額（1円未満は切り捨てる）を支給する。賃貸住宅以外の場合は1万円を支給する。

＜設問１＞　次の「通勤情報」ワークシートの作成に関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

自家用車や自転車を利用した通勤は距離に応じて支給額が変わることから，A君は「自家用情報」ワークシートを作成した。なお，1000kmを超える通勤をする社員はいない。

	A	B	C	D
1	距離(Km)		支給額	
2	最低	最大	自家用車	自転車
3	0	1.9	0	0
4	2	9.9	4,000	2,000
5	10	14.9	7,000	4,000
6	15	24.9	12,000	7,000
7	25	34.9	18,000	10,000
8	35	1000	24,000	14,000

図１　「自家用情報」ワークシート

次に，「通勤情報」ワークシートを作成した。このワークシートは，「自家用情報」ワークシートを参照する。なお，社員番号は全員分を正しく入力した。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		経路１		経路２		経路３		小計１	小計２	小計３	合計		種別リスト
2	社員番号	種別	金額／距離	種別	金額／距離	種別	金額／距離						公共交通
3	101002	自転車	3	公共交通	30,000	公共交通	70,000	2,000	30,000	70,000	102,000		自家用車
4	101003	自家用車	10	公共交通	20,000			7,000	20,000		27,000		自転車
5	124002	自転車	4					2,000			2,000		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		
20	402332	公共交通	12,000					12,000			12,000		
21	404921	公共交通	8,000					8,000			8,000		
22	501231	公共交通	32,000					32,000			32,000		

図２　「通勤情報」ワークシート

「通勤情報」ワークシートは，全部で３つの経路が入力できるようになっている。

小計１は経路１の，小計２は経路２の，小計３は経路３の金額を表示するが，該当する経路に情報が入力されていない場合は空表示とする。

・B列，D列，F列の種別は，電車やバスを使った"公共交通"，"自家用車"，"自転車"の３つから入力する。利用する経路が無い場合は空欄にする。

・C列，E列，G列の金額／距離は，"公共交通"の場合は１カ月の定期代，それ以外は100m単位に切り上げた距離をkmに換算して入力する。利用する経路が無い場合は空欄にする。

・セル H3 に経路１の金額を表示する。距離が同じでも種別が″自家用車″と″自転車″では支給額が異なるため，M 列の種別リストを MATCH 関数で利用し，支給額を取り出す列位置を変化させる。セル H3 に次の式を入力し，セル H4～H22 に複写する。

= IF(B3=″″, ″″, IF((1) , C3, (2)))

同様の式をセルI3とJ3に入力し，それぞれセルI4～I22およびJ4～J22に複写する。

・セルK3にセルH3～J3の合計を求める式を入力し，セルK4～K22に複写する。

(1) の解答群

ア．B3=″公共交通″　　イ．B3<>″公共交通″

ウ．B3=″自家用車″　　エ．B3<>″自家用車″

(2) の解答群

ア．VLOOKUP(C3, 自家用情報!A$3:D$8, MATCH(B3, M$2:M$4, 0), 1)

イ．VLOOKUP(C3, 自家用情報!A$3:D$8, MATCH(B3, M$2:M$4, 0)+1, 1)

ウ．VLOOKUP(C3, 自家用情報!B$3:D$8, MATCH(B3, M$2:M$4, 0)-1, 1)

エ．VLOOKUP(C3, 自家用情報!B$3:D$8, MATCH(B3, M$2:M$4, 0), 1)

＜設問２＞　次の「給与情報」ワークシートの作成に関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

社員の給与に関する情報を作成するため「給与情報」ワークシートを作成する。

	A	B	C	D	E	F	G	H	I	J	K	L
1	社員番号	氏名	配偶者 0:なし 1:あり	子どもの人数	役職	家賃	基本給	役職手当	家族手当	通勤手当	住宅手当	合計
2	101002	滝 洋介	1	4	部長	0	300,000	50,000	45,000	100,000	10,000	505,000
3	101003	志賀 基祐	1	2	課長	120,000	240,000	30,000	35,000	27,000	30,000	362,000
4	124002	若槻 敏也	0			70,000	180,000	0	0	2,000	30,000	212,000
5	126008	樋口 博明	0	0	次長	0	280,000	40,000	0	24,800	10,000	354,800
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
19	402332	松谷 莉沙	1	2		64,000	160,000	0	35,000	12,000	30,000	237,000
20	404921	竹内 結衣	0	0	主任	80,000	210,000	10,000	0	8,000	30,000	258,000
21	501231	日比野 恵梨香	0	0		78,000	180,000	0	0	32,000	30,000	242,000

図３　「給与情報」ワークシート

・A列に社員番号，B列に社員の氏名，C列に配偶者の有無を0と1で，D列に扶養している子どもの人数を入力した。A列の社員番号は，「通勤情報」ワークシートのA列に入力した社員番号と一致している。

・E列の役職は，″部長″，″次長″，″課長″，″係長″，″主任″の中から入力する。役職が無い社員は空欄にする。

・F列の家賃は，賃貸住宅に居住している場合は家賃，そうでない場合は0を入力する。

・G列は基本給を入力する。

・H列の役職手当は，「役職手当」ワークシートを検索して表示する。ただし，役職が無い場合は0を表示する。

	A	B
1	役職名	金額
2	部長	50,000
3	次長	40,000
4	課長	30,000
5	係長	20,000
6	主任	10,000

図４　「役職手当」ワークシート

セルH2に次の式を入力し，セルH3～H21に複写する。

= [(3)]

・I列の家族手当は，家族手当の規則から，配偶者は1万5千円，子どもの場合，2人目までは1人につき1万円，3人目以降は1人につき5千円を支給するので，セルI2に次の式を入力し，セルI3～I21に複写した。

= C2*15000 + IF(D2<=2, [(4)], [(5)])

・J列の通勤手当は，図２の「通勤情報」ワークシートから検索する。上限が10万円であることから次の式をセルJ2に入力し，セルJ3～J21まで複写した。

= [(6)]

・K列の住宅手当は，住宅手当の規則から，賃貸住宅に居住している場合は3万円を上限として家賃の半額を支給し，それ以外は1万円を支給するので，セルK2に次の式を入力し，セルK3～K21まで複写した。

= IF([(7)], 10000, [(8)])

(3) の解答群

ア．IF(E2="", 0, VLOOKUP(A2, 役職手当!A$2:B$6, 2, 0))
イ．IF(E2="", 0, VLOOKUP(E2, 役職手当!A$2:B$6, 2, 0))
ウ．VLOOKUP(A2, 役職手当!A$2:B$6, 2, 0)
エ．VLOOKUP(E2, 役職手当!A$2:B$6, 2, 0)

(4)，(5) の解答群

ア．D2*5000　　イ．D2*5000+20000　　ウ．D2*10000
エ．D2*10000+5000　　オ．(D2-2)*5000　　カ．(D2-2)*5000+20000

(6) の解答群

ア. IF(VLOOKUP(A2, 通勤情報!A$3:K$22, 11, 0)>100000, 100000, 通勤情報!A3)

イ. MAX(VLOOKUP(A2, 通勤情報!A$3:K$22, 11, 0), 100000)

ウ. MIN(VLOOKUP(A2, 通勤情報!A$3:K$22, 11, 0), 100000)

エ. VLOOKUP(A2, 通勤情報!A$3:K$22, 11, 0)

(7) の解答群

ア. F2=0

イ. F2>10000

ウ. F2>30000

エ. F2<30000

(8) の解答群

ア. MAX(ROUNDDOWN(F2/2, 0), 30000)

イ. MIN(ROUNDDOWN(F2/2, 0), 30000)

ウ. ROUNDDOWN(F2/2, 0)

エ. ROUNDDOWN(F2/2, 0) - 30000

選択問題　アセンブラの問題

次のアセンブラ言語CASLⅡプログラムの説明を読み，各設問に答えよ。

［プログラムの説明］

パラメタで与えられたデータを左または右へ循環シフトする副プログラム JUNSF である。

JUNSF は，図１のような形式でパラメタが格納された先頭番地を GR1 に設定して呼び出される。二つ目のパラメタに'L'と'R'以外の文字は設定されない。また，三つ目のパラメタは正数として与えられる。

アドレス	
(GR1) + 0	循環シフトの対象データ
+ 1	左シフト('L')または右シフト('R')
+ 2	循環シフトするビット数
+ 3	循環シフトの結果

図１　パラメタの構造

通常の論理シフトは，シフトによりあふれたビットは無くなり，空いた領域に 0 が格納されるが，循環シフトでは，シフトによりあふれたビットが空いた領域に格納される（図２）。

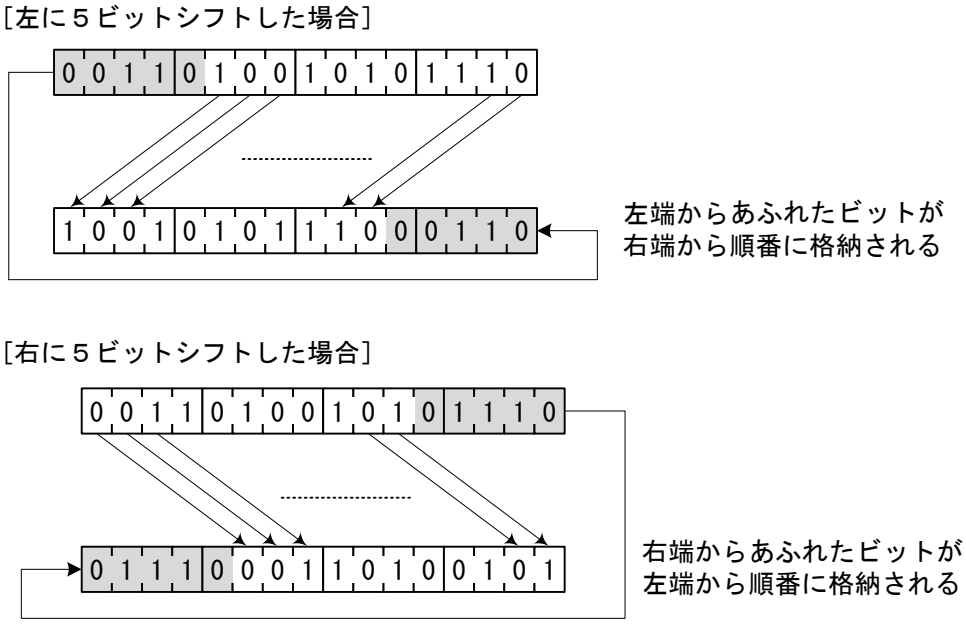


図２　循環シフトの例

［プログラム］

行番号	ラベル	命令	オペランド	コメント
100	JUNSF	START		
110		RPUSH		
120		LD	GR2,2,GR1	;シフトするビット数を取り出す
130		ST	GR2,M	
140		(1)		
150		SUBA	GR3,M	
160		LD	GR5,0,GR1	;シフトするデータを取り出す
170		LD	GR6,GR5	
180		LD	GR0,1,GR1	;左シフトか右シフトか取り出す
190		CPL	GR0,='L'	
200		JZE	L1	
210		(2)		
220		(3)		
230		JUMP	L2	
240	L1	SLL	GR5,0,GR2	
250		SRL	GR6,0,GR3	
260	L2	OR	GR5,GR6	
270		(4)		；シフト結果を格納する
280		RPOP		
290		RET		
300	M	DS	1	
310		END		

＜設問１＞　プログラム中の［　　］に入れるべき適切な字句を解答群から選べ。

(1) の解答群

ア．LD　GR3,=0　　イ．LD　GR3,=15

ウ．LD　GR3,=16　　エ．LD　GR3,M

(2) の解答群

ア．SLL　GR5,0,GR2　　イ．SLL　GR5,0,GR3

ウ．SLL　GR5,1,GR2　　エ．SLL　GR5,1,GR3

(3) の解答群

ア．SRL　GR6,0,GR2　　イ．SRL　GR6,0,GR3

ウ．SRL　GR6,1,GR2　　エ．SRL　GR6,1,GR3

(4) の解答群

ア．**ST　GR5,3,GR1**　　イ．**ST　GR5,3,GR2**

ウ．**ST　GR6,3,GR1**　　エ．**ST　GR6,3,GR2**

＜設問２＞　プログラムの修正に関する記述中の［　　］に入れるべき適切な字句を解答群から選べ。

現在のプログラムは，三番目のパラメタ(循環シフトするビット数)に設定するビット数の範囲が［　(5)　］であれば正しく処理され，それ以外の数値が設定された場合，結果は［　(6)　］となる。

循環シフトは，16 ビットシフトさせると一巡して元にもどる。そこで，シフトするビット数が正の値であれば［　(5)　］の範囲外の値が設定されても正しい循環シフトの結果が得られるように，行番号 120 と 130 の間に 1 命令を挿入した。

行番号	ラベル	命令	オペランド
120		LD	GR2,2,GR1
125		(7)	
130		ST	GR2,M

図３　修正を含む命令群

(5) の解答群

ア．**0～15**　　イ．**0～16**　　ウ．**1～15**　　エ．**1～16**

(6) の解答群

ア．すべてのビットが 0　　イ．すべてのビットが 1

ウ．元データと全く同じ　　エ．元データの２の補数

(7) の解答群

ア．**ADDA GR2,=#000F**　　イ．**AND GR2,=#000F**

ウ．**OR GR2,=#000F**　　エ．**SUBA GR2,=#000F**