

受験番号	志望学科・コース
	学科
	コース

〔 計・ソ専門 - 1 〕

問題 1

次の並び替えを行う手続き s に関して以下の問 (1) ~ (4) に答えよ。

整数型の定数 n が非局所的に定義されて、添字が 0 から n までの配列 a が宣言されているものとする。配列 a の配列要素のデータ型は実数型とする。添字 j ($0 \leq j \leq n$) の配列要素は $a[j]$ でアクセスできる。

実数型の相異なる n 個の値が、配列 a の添字 1 から n までの部分に入力されているものとして、その n 個のデータを値が小さい順に並べ替えて、配列 a の添字 1 から n までの部分に格納しなおすことを考える。ただし、 $a[0]$ はどのように値が変化してもよいものとする。

- (1) 手続き s が正しく働くように および を適切にうめて、2つの代入文を完成せよ。ただし、文 $x:=e$ は代入文を表し、式 e の評価値を変数 x に代入することを表す。
- (2) のところの代入文が存在しないと、手続き s は一般には正しく動かない。その理由を説明せよ。
- (3) while文での実数型の値の大小比較 $d < a[j-1]$ が、手続き s が終了するまでに何度実行されるかについて考える。この実行回数は、配列 a の添字 1 から n までの部分に入力された値の大小の関係により、一般に異なる。次の2つの場合について実行回数を答えよ。ただし、答は n を用いた正確な式で表せ。オーダ表記は用いないこと。
 - ・実行回数が最も多い場合
 - ・実行回数が最も少ない場合
- (4) 上の問 (3) で、実行回数が最も少ない場合は、入力データがどのような関係にある場合かを説明せよ。

```

procedure s;
  var i, j : integer;
      d : real;
begin
  for i:=2 to n do begin
    d :=  ;
     := d;
    j:=i;
    while d < a[j-1] do begin
      a[j] := a[j-1];
      j:=j-1;
    end;
    a[j] := d;
  end;
end;

```

受験番号	志望学科・コース
	学科
	コース

[計・ソ専門-2]

問題 2

(1) 表1の状態遷移表で表されている順序機械(入力 x) を、同期式順序回路として設計することを考える。ただし、この設計では、次の3項目を仮定している。

- 2つのDフリップフロップを使うこととし、一方のDフリップフロップの入力と出力をそれぞれ Y_1 と y_1 、もう一方のDフリップフロップの入力と出力をそれぞれ Y_2 と y_2 とする。
- 順序回路の出力を Z とする。
- 状態 S_0, S_1, S_2, S_3 へは、それぞれ 00, 01, 10, 11 を割り当てるとする。

このとき、 Y_1, Y_2 および Z に対応する論理関数を、 x, y_1, y_2 を論理変数とする論理式で表せ。(ヒント: Dフリップフロップの動作表を表2に示す。表2では例えば、時刻 t において、 $D=1$ であれば、時刻 $t+1$ での出力の値 $Q(t+1)$ は1になることを示している。)

表1: 状態遷移表

現状態	次状態、出力	
	入力 x	
	0	1
S_0	$S_0, 0$	$S_1, 0$
S_1	$S_2, 0$	$S_1, 0$
S_2	$S_0, 0$	$S_3, 0$
S_3	$S_2, 1$	$S_1, 0$

表2: Dフリップフロップの動作表

D	$Q(t+1)$
0	0
1	1

(2) 3つの入力 (x_1, x_2, x_3) と1つの出力を持つ組合せ回路 F (図1の破線部) を考える。 F の内部には、3つの入力 (x_1, x_2, x_3) と1つの出力を持つ組合せ回路 H と、4つの入力 (x_1, x_2, x_3, w) と1つの出力を持つ組合せ回路 G が含まれており、図1に示したように、 H の出力が G の入力 w に結線されている。ここで、 H は表3の真理値表で表現される論理関数 h を実現しており、 G は表4の真理値表で表現される論理関数 g を実現しているとする。

表4: 論理関数 g

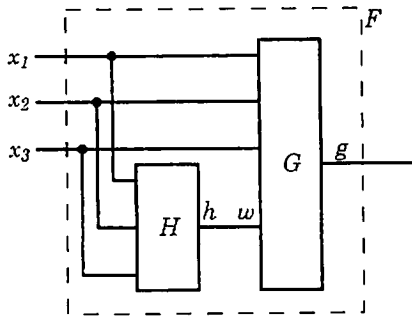


表3: 論理関数 h

x_1	x_2	x_3	h
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

x_1	x_2	x_3	w	g
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

図1: 組合せ回路 F

- 入力 (x_1, x_2, x_3) の値の組合せによっては、 w の値に依存せずに、 G の出力が決まる場合がある。例えば、 $(x_1, x_2, x_3) = (0, 0, 0)$ であれば、 w の値に依存せずに、 G の出力は1になる。このような入力 (x_1, x_2, x_3) の値の組合せを全て示せ。
- 組合せ回路 G が実現している論理関数 g の最簡積和形表現を求めよ。
- F が実現する関数を変えないという条件のもとで、 H が実現する論理関数 h を h' に変更する。 h に対する最簡積和形表現と比べて、 h' に対する最簡積和形表現に含まれる積項の数を少なくするようにしたい。(a) の結果を参考にして、変更後の論理関数 h' を1つ示し、その最簡積和形表現を求めよ。

受験番号	志望学科・コース
	学科
	コース

問題 3

右下の図は中央処理装置と主記憶装置からなるシステムを表している。図には、主な構成要素とデータの流れを表す矢印が描かれている。

この中央処理装置の命令は、命令の種類とオペランドアドレスから構成される。中央処理装置は、プログラムカウンタで指定される主記憶装置のアドレスの内容を命令とみなし、以下の命令の種類に従い処理を実行する。ある命令の実行を開始した後、その命令の実行が終了するまで次の命令を実行しない。また、演算回路は加減算及び2つの入力的一方を出力する機能があり、プログラムカウンタはその値を1増やす機能がある。演算回路およびプログラムカウンタにおいて、演算結果のオーバフローは起きないものとする。以下では、データを10進数で表す。

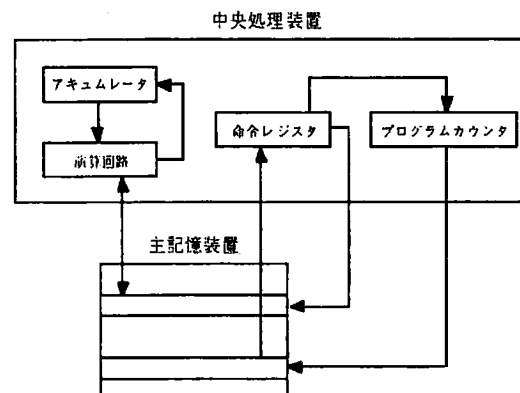


図 中央処理装置と主記憶装置からなるシステム

- ・命令の種類がLOADのとき、その命令のオペランドアドレスで指定された主記憶装置のアドレスの内容をアキュムレータに転送し、プログラムカウンタの値を1増やす。
- ・命令の種類がSTAのとき、アキュムレータの内容をその命令のオペランドアドレスで指定された主記憶装置のアドレスに書き込み、プログラムカウンタの値を1増やす。
- ・命令の種類がADDのとき、アキュムレータの内容に、その命令のオペランドアドレスで指定された主記憶装置のアドレスの内容を加えた値をアキュムレータに転送し、プログラムカウンタの値を1増やす。
- ・命令の種類がSUBのとき、アキュムレータの内容から、その命令のオペランドアドレスで指定された主記憶装置のアドレスの内容を引いた値をアキュムレータに転送し、プログラムカウンタの値を1増やす。
- ・命令の種類がJMPのとき、アキュムレータの値が0でないときその命令のオペランドアドレスの値をプログラムカウンタに転送する。アキュムレータの値が0のとき、プログラムカウンタの値を1増やす。
- ・命令の種類がSTOPのとき、それ以降命令実行を行わない。

(1) プログラムカウンタで指定される主記憶装置のアドレスの内容が以下の場合、それぞれについてシステムの処理の実行手順を、データがどこからどこに転送されるかがわかるように具体的に述べよ。以下は実行手順の例である。

例: 命令の種類がLOADで、オペランドアドレスが400の場合

手順1: プログラムカウンタで指定される主記憶装置のアドレスの内容を命令レジスタに転送する。

手順2: 命令の種類を判定する。この場合LOADであると判定する。

手順3: 主記憶装置の400番地の値を取り出し、演算装置に転送する。

手順4: 演算装置は主記憶装置からの値を出力し、その出力をアキュムレータに転送する。

手順5: プログラムカウンタの値を1増やす。

(a) 命令の種類がADDで、オペランドアドレスが200の場合

(b) 命令の種類がJMPで、オペランドアドレスが100かつアキュムレータの値が20の場合

(2) メモリの100番地にあらかじめ正の数値が格納されているものとする。1からこの数値までの和を求め、101番地目に計算結果を格納する命令系列を求めよ。101および102番地目の値はあらかじめ、それぞれ0および1になっているものとし、命令系列の実行終了後100番地目の値はどのような値になってもよいものとする。また、命令系列を開始するときのプログラムカウンタの値は0とする。以下の例は、5番地に格納されている命令の種類がSTA、オペランドアドレスが100であるような命令の例である。

5: STA : 100

命令系列は、番地の小さいほうから順に一行一命令となるように解答すること。