2019年度
ＡＯ入試１次選考問題
情報理工学部
情報科目

注 意 事 項

1. 試験開始の合図があるまで、問題用紙を開いてはいけません。
2. 解答はすべて、所定の解答用紙に記入してください。
3. 解答用紙に受験番号と氏名（フリガナ）を記入してください。
4. 解答時間は６０分です。問題は１３ページあります。
5. 問題用紙・解答用紙および計算用紙はすべて回収します。一切持ち帰ってはいけません。

〔Ⅰ〕 次の設問（A）と設問（B）に答えなさい。

設問（A）

次の文章について、空欄a～dを埋める適切な言葉を選択肢の中から選び、記号で答えなさい。

インターネット上の通信は、基本的に［ a ］と呼ばれる通信規約を元にして行われており、その通信規約に従う機器には、［ b ］が付与されている。インターネットの初期の頃から使われてきている［ b ］は［ c ］と呼ばれるものであり、［ d ］で表現されるアドレスとなっている。近年はインターネットに接続される機器の数が増えてきて［ c ］の［ d ］ではアドレスを表現するには足りない状況になっており、128bit で表現される IPv6 のアドレスへ移行を急ぐよう世界中で言われ続けている。

設問（A）a ～ d の選択肢

ア. 32bit	イ. 64bit	ウ. 256bit	エ. 512bit	オ. IP (Internet Protocol)
カ. HTTP (Hyper Text Transfer Protocol)		キ. FTP (File Transfer Protocol)		
ク. IPv4	ケ. IPv5	コ. IP アドレス		

設問（B）

一般的なコンピュータの画面は、ピクセル（画素）と呼ばれる点の集まりとして構成されている。このピクセルの色を、赤（R）、緑（G）、青（B）の光の三原色として表す場合、それぞれ8bit ずつで表現すると 1 ピクセルあたり 24bit を用いることになる。コンピュータ内部では、画面表示するピクセルの色の情報を画面描画用メモリに保持することが多い。仮に、4K ディスプレイ（横 3840 ピクセル×縦 2160 ピクセル）の画面を RGB の 24bit で表示しようとすれば、画面描画用メモリは約何 MB 必要になるか、次の選択肢から選んで記号で答えなさい。

設問（B）の選択肢

ア. 約 18MB	イ. 約 25MB	ウ. 約 33MB	エ. 約 40MB

〔Ⅱ〕 以下の文章を読んで、下の設問（A）～ 設問（C）に答えなさい。

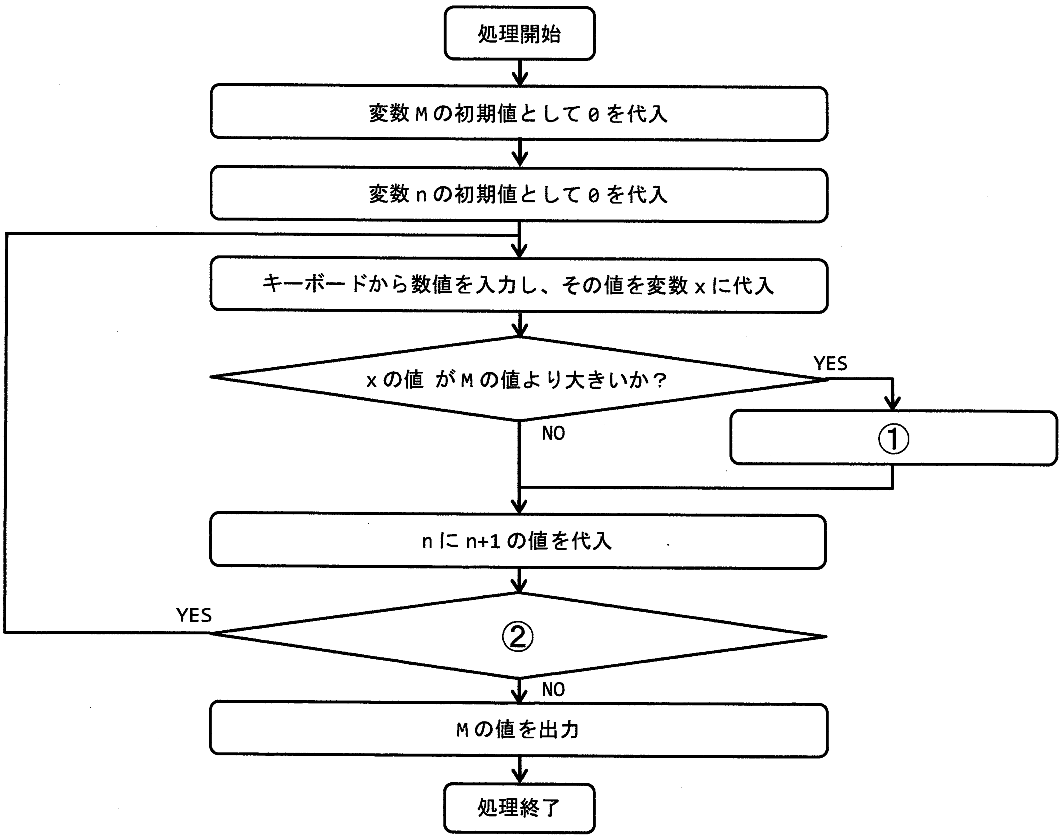
符号付きの数値をキーボードから 1 つずつ入力し、10 個続けて入力した後に入力数値の最大値を出力するプログラムを作成したい。プログラムでは、符号付き数値を 1 つだけ格納できる変数を複数個用いることができる。変数にはキーボードから入力した数値や任意の定数を代入したり、変数の値に対する四則演算結果を代入することができる。また、変数の数値が定数や他の変数の数値より大きいか小さいかの判定を行い、この判定結果に応じてプログラムの処理の流れを分岐させることができる。

設問（A）

プログラム内では最低何個の変数を用いる必要があるか答えなさい。また、それぞれの変数のプログラム内での役割を述べなさい。

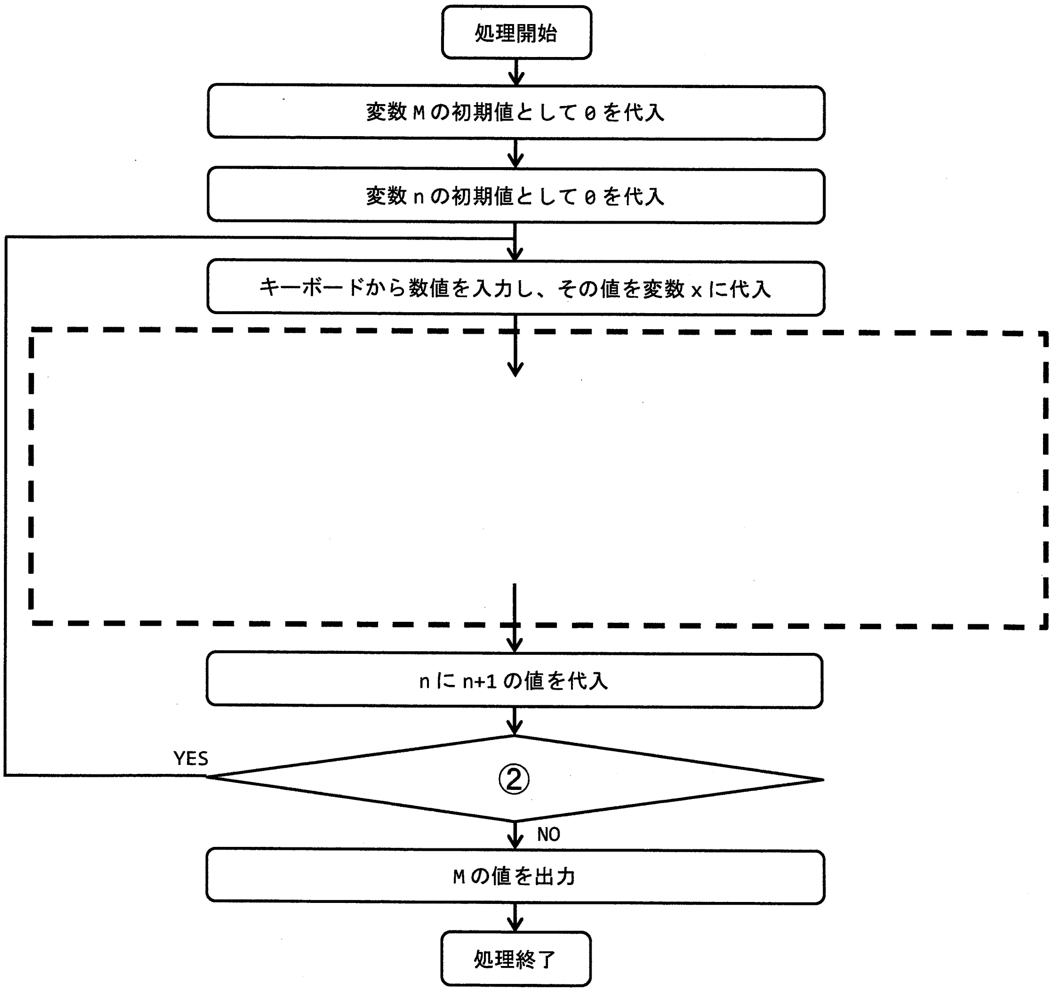
設問（B）

プログラムのフローチャートを下のように考えた。このフローチャートの空白部分①と②に入る処理をそれぞれ答えなさい。



設問（C）

設問（B）のフローチャートは、入力値がある条件にあてはまる場合には正しい最大値を出力できない。どのような場合にどのような誤った出力を行うか答えなさい。また、この誤りが生じないようにフローチャートの点線四角部分を修正することにした。修正後の内容を解答用紙の点線四角内に書き入れなさい。ただし、フローチャートの②には設問（B）の②と同一の処理内容が入る。

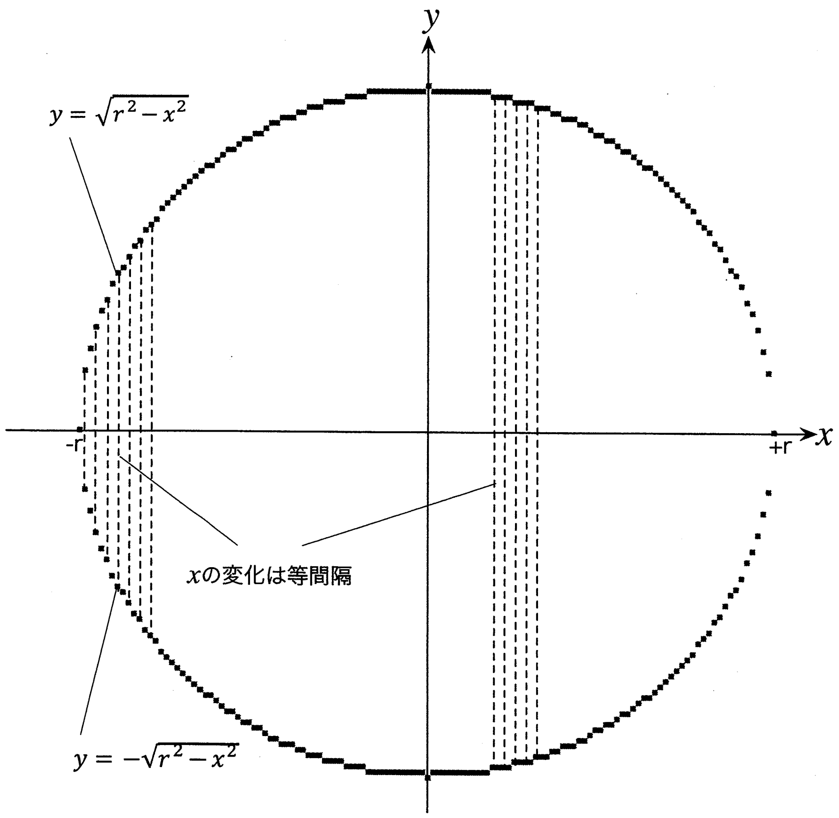


〔Ⅲ〕 次の文章を読んで、設問（A）と設問（B）に答えなさい。

コンピュータの二次元画面上に円周を描く処理を考える。ここでは、画面上に何か表示させるのに使える命令は、背景色とは違う色の点を打つ命令（x, y 座標を整数値で指定）のみとし、その打った点によって円周に近似した図形を描くこととする。なお、本問題での点とは、画面を構成する四角い画素のことを指す。以下、画面上に点を打つ座標を計算する方法について考える問題として、以下の設問について答えなさい。

設問（A）

まず、円の方程式　$x^2 + y^2 = r^2$　（r：半径）を元に、x 座標を $-r$ から r まで 1 ずつ増やし、下図のように y 座標の正負両方の値を計算しつつ点を打っていく方法を考えた。その際、y 座標の計算では平方根の演算によって実数が得られるが、実際に点を打つ際には小数点以下は切り落として整数値として用いる。ただ、この方法では下図のように点と点の間に隙間ができることがある。このような隙間をなくすためには、この描画処理の方法にどのような改良や工夫が可能かを答えなさい。



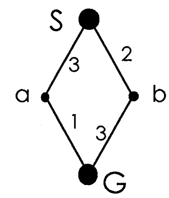
x 座標を $-r$ から r へ変化させつつ y 座標を求めて打点して円を描く処理の概念図

設問（B）

上記の設問（A）の方法とは別の方法で、円周を近似した図形を描くこともできる。その方法について概要を説明しなさい（図を描いて説明してもよい）。また、その手法を用いる際の注意点があれば、それについても説明しなさい。

〔IV〕 以下の文章を読んで、設問（A）～ 設問（D）に答えなさい。

ある地点から別の地点までの時間や空間的に最も短い経路を求めることはカーナビや乗り換え案内などで幅広く利用されている。本問では、与えられた経路図から開始地点から目的地までの経路をもれなく求め、求めた一つ一つの経路の長さを計算する単純な方法を使って、最短経路とその長さを探索する方法について考える。本問の探索方法は2つのステップ、(Step1)経路図を木構造に置き換え、(Step2)木構造を探索して最短経路とその長さを求める、から構成されている。図1-Aと図1-Bを例に探索方法を説明する。なお探索方法の説明の後に、4つの設問を記述している。



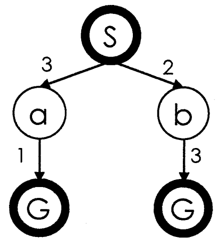


図.1-A 経路図　　図.1-B 経路図の木構造

図1-Aは経路図であり、SとG, a, bは交点である。交点と交点の間の線は交点間に経路があることを示し、線の数字はその経路の長さを示している。交点Sは開始地点を、交点Gは目的地をそれぞれ示す。例えば、交点Sと交点aの間には経路があり、その経路の長さは3である。一方交点Sと交点Gの間には線がないため、直接の経路はないことを示している。

次に(Step1)の結果、図1-Aの経路図を置き換えた木構造を図1-Bに示す。木構造では、経路図の交点を木構造の節(木構造では○)に、経路を枝(木構造では→)で示す。木構造の節と節の間に、枝がある場合は経路があることを示し、枝がない場合は経路がないことを示す。枝には重みと呼ばれる数字が付いている。この数字は経路図の交点間の長さの値である。開始地点Sからは、長さ3の交点aへの経路と長さ2の交点bへの経路がある。そのため、木構造の節Sは節aと節bの間に枝があり、節aへの枝の重みは3であり、節bへの枝の重みは2である。特に、経路図の交点Sに対する木構造の節を根といい、経路図の交点Gに対する木構造の節を葉と呼ぶ。この木構造の根からそれぞれの葉までの節の順序は、経路図の開始地点から目的地までのそれぞれの交点の順序を示す。

最後に(Step2)で、この木構造を用いて、最短経路の経路とその長さを求める。求める具体的な手順を図2[手順1]に示す。図1-Bと[手順1]を用いると、図1-Aの経路図において、経路数

は、葉の数でもある、2 通り存在し、それぞれの経路の長さ t はそれぞれ 4 と 5 であり、最終的な最短経路は、(S, a, G)であり、その経路の長さは 4 と得られる。

1. 最短経路を p とし、現在の最短経路の長さ m を無限大とし、すべての葉の集合 L を作る。
2. 葉の集合 L が、空(なにもない)でなければ 3. へ進み、空であれば 7. へ進む。
3. 葉の集合 L から木構造において一番左にある葉を一つ取りだし、取り出した葉を ℓ とする。葉の集合 L から取り出した葉 ℓ を削除する。
4. 根から、取り出した葉 ℓ までの経路上のすべての枝の重みを加算する。得られた値を根からその葉 ℓ への経路の長さ t とする。
5. もし求めた経路の長さ t が現在の最短経路の長さ m より短ければ、求めた経路の長さ t に最短経路の長さ m を更新し、根から葉 ℓ までの経路を p に保存する。
6. 2. へ戻る。
7. 保存された経路 p が最短経路であり、現在の最短経路の長さ m が最短経路の長さである。

図.2 [手順 1]

この 2 つのステップからなる探索手法を適用し、図 3-A の経路図の最短経路とその長さを求める。まず(Step1)で経路図を木構造に置き換えた。図 3-B に木構造を示す。図 3-B の木構造に対して、[手順 1]を適用し、最短経路とその長さを求めてゆく。次ページの 4 つの設問に答えなさい。

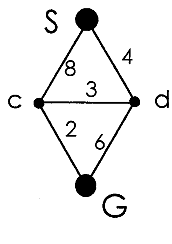


図.3-A 経路図

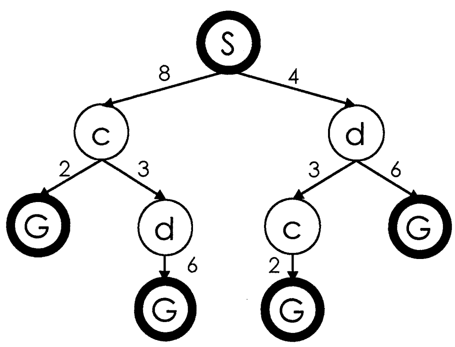


図.3-B 木構造

設問（A）

図 3-A 経路図中の開始地点(S)から目的地(G)までの経路は何通り存在するかを答えなさい。ただし、一つの経路において、同じ交点は2度以上通過しないものとする。

設問（B）

図 3-A 経路図中の開始地点(S)から目的地(G)までの最終的な最短経路とその長さを記述例に従って答えなさい。

図 1-A の経路図に対する最短経路と長さの記述例: (S, a, G), 4

設問（C）

[手順 1]において、図 3-B の木構造に対して、変数 m の値はどのように変化したかを記述例に従って答えなさい。

図 1-B の木構造に対する記述例: 無限大 → 4

設問（D）

図 3-A より大きな経路図に対して[手順 1]を適用すると、木構造の根からすべての葉に対する経路を調べるため、[手順 1]は探索の効率が悪い。木構造を用いて、最短経路とその長さを効率良く求めるために、[手順 1]をどのような処理に改善すればよいかについて考える。

(1)　[手順 1]の効率が悪い箇所を示しなさい。

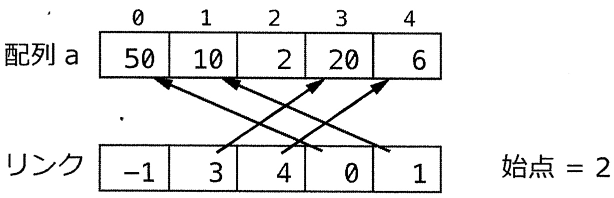
(2)　(1)で指摘した箇所を改善する手順を答えなさい。

（白紙）

〔V〕 以下の文章を読んで、設問（A）～ 設問（C）に答えなさい。なお、この問いでは問題冊子の最後に示す記法を使ってプログラムを記述する。

N 個のデータが配列に格納されているとする。各データごとに、そのデータの次に大きいデータがどの位置にあるかを示す添字を用意し、これらをリンクと呼ぶ。ただし、最も大きいデータの持つリンクは特別に −1 を値とする。また、最も小さいデータが格納されている位置の添字を始点と呼ぶ。

たとえば、5つの整数が以下の順番に配列 `a` に格納されていた場合の、各要素に対応するリンクを示す。また、始点は 2 である。



このようなデータ構造を使うと、元のデータの順序を変更せずにデータ間の順序関係を示すことができる。実際のプログラムでも、大規模なデータや、変更することが好ましくないデータを扱う場合などに利用される。

設問（A）

要素数 N の配列 `a` のうち、先頭の `m` 個（0 < `m` < N）だけに意味のあるデータが格納されているとする。配列 `link` にも先頭の `m` 個に対応するリンクが格納され、変数 `start` には `m` 個のデータのうちの最も小さいものの添字（すなわち始点）が格納されている。このとき、配列 `a` に格納された `m` 個のデータを小さい順にすべて表示するプログラムを記述しなさい。ただし、整数を格納する変数として `i`, `j` を利用してよい（すべてを利用する必要はない）。

設問（B）

設問（A）と同様、要素数 N の配列 `a` の先頭の `m` 個（0 < `m` < N）にデータが格納され、配列 `link` と変数 `start` には対応するリンクと始点が格納されているとする。いま、`a[m]` に新しいデータが追加されたとして、リンクと始点を更新したい。ただし、配列 `a` に格納される整数値はすべて互いに異なるとする。

たとえば、上記の例で配列 `a` に 9 が追加された場合、リンクと始点は次のようになる。

	0	1	2	3	4	5	
a	50	10	2	20	6	9	
link	-1	3	4	0	5	1	start = 2

また、9 ではなく、1 が追加された場合、リンクと始点は次のようになる。

	0	1	2	3	4	5	
a	50	10	2	20	6	1	
link	-1	3	4	0	1	2	start = 5

さらに、追加されたデータが 80 だった場合、リンクと始点は次のようになる。

	0	1	2	3	4	5	
a	50	10	2	20	6	80	
link	5	3	4	0	1	-1	start = 2

では、追加されたデータが 12 だった場合、リンクと始点はどのように更新されるか、上の例にならって答えなさい。

設問（C）

設問（B）で論じたように、配列 `a` の `m` 個のデータについて、配列 `link` と変数 `start` の値が設定されていると仮定し、このとき、`a[m]` に新しく追加されたデータに応じて配列 `link` と変数 `start` の値を更新するプログラムを記述しなさい。ただし、必要に応じて変数 `i, j, k` を利用してよい（すべてを利用する必要はない）。

プログラムの記法の説明

問〔V〕でプログラムを記述するために用いる記法について説明する。

文

変数 = 式	変数に式の値を代入する。以下の説明ではこの文を「代入文」と呼ぶ。
for (代入文１; 条件式; 代入文２) ... end	まず、代入文１を実行し、条件式を評価する。条件式が偽であれば何もしない。条件式が真のとき、end までの命令を実行し、次に代入文２を実行する。その後、条件式が真である間、end までの命令と代入文２を実行する。
while 条件式 ... end	条件式が真である間、end までの命令を実行する。

if 条件式
  ...
else
  ...
end
	条件式が真のとき、else までの命令を実行し、偽のとき、else から end までの命令を実行する。else を記述しない場合、条件式が真のときに end までの命令を実行し、条件式が偽のときは何もしない。

break	for 文、または while 文（これらをループ文と呼ぶ）の内部でのみ利用できる。実行すると繰り返しの処理を打ち切り、ループ文の次の文の実行に移る。ループ文の中でループ文が使われている（ネストしている）時は、一番内側のループ文の処理だけが打ち切られる。
print(式)	式の値を表示する。

配列

配列名[式]	含まれる要素の数があらかじめ決められた配列を利用できる。 配列 a の要素数が N（N は正の整数）のとき、配列 a は 0 番目から（N−1）番目までの要素を持ち、i 番目の要素は a[i] という記法で表現する。

比較演算子

A == B	A と B の値が等しい。
A != B	A と B の値が等しくない。
A <= B	A の値が B の値以下である。
A < B	A の値が B の値より小さい。
A >= B	A の値が B の値以上である。
A > B	A の値が B の値より大きい。

算術演算子

+	加算（足し算）を行う。
-	減算（引き算）を行う。
*	乗算（掛け算）を行う。
/	除算（割り算）を行う。ただし、結果は実数で表される。

複数の算術演算子が混在した式では * と / の計算が + や - よりも先に行われる。
算術演算子と比較演算子が混在した式では、算術演算子が先に計算される。
また、式の中で( ) を使い、計算の順序を示すことができる。

プログラムの例

(1) 1 + 2 + 3 + … と加算を繰り返して、その値を表示する。合計が 100 を超えたら終わる。

```
s = 0
i = 1
while s <= 100
    s = s + i
    print(s)
    i = i + 1
end
```

(2) 与えられた要素数 n の配列 a の内容を、同じ大きさの配列 b にコピーする。ただし、配列 a の要素で負の数があれば、b には代わりに 0 を代入する。

```
for (i = 0; i < n; i = i + 1)
    if a[i] >= 0
        b[i] = a[i]
    else
        b[i] = 0
    end
end
```