

受験番号	
------	--

令和2年度
大分工業高等専門学校編入学試験問題

専門科目 I
(情報工学科)

令和元年11月16日(土)
9:00 ~ 10:20

【注意事項】

1. 指示があるまで問題用紙は開いてはいけません
2. この問題は表紙のほかに5ページあります
3. 全てのページの受験番号欄に受験番号を記入しなさい
4. 各解答はその問題の所定の欄に記入しなさい

1. リスト1は、点数が高い順に順位付けするC言語プログラムである。

66点, 56点, 78点, 55点, 88点, 90点, 56点, 76点, 73点, 58点

と10人分の点数があったときに、たとえば66点の順位は、66点より大きい点が見つかるたびにその数を数えて、5回数え上げる(初期値は1)ことにより、6位と分かる。

(1) リスト1のプログラム中の①に該当する処理を解答欄に書きなさい。

(2) 実行結果として出力される内容を解答欄に書きなさい。

```
#include <stdio.h>
#define Num 10

int main(void)
{
    int a[] = {66, 56, 78, 55, 88, 90, 56, 76, 73, 58};
    int i, j, juni[Num];

    for (i = 0; i < Num; i++)
        juni[i] = 0;           // 配列の初期化

    for (i = 0; i < Num; i++){
        juni[i]=1;
        for (j = 0; j < Num; j++){
            if (a[j] > a[i])
                ①;
        }
    }

    printf(" 得点 順位¥n");
    for (i = 0; i < Num; i++)
        printf("%d  %d¥n", a[i], juni[i]); // 順位の出力
    return 0;
}
```

(1)	①
(2)	得点 順位

リスト1

2. 前問1のプログラムはデータが n 個の場合、繰り返し回数は n^2 となるため、データ数が増えると計算量が膨大になる。そこで、リスト2のアルゴリズムに従い、プログラムを改良する。リスト3のプログラム中の②~④に該当する処理を次のページの解答欄に書きなさい。

順位付けアルゴリズムの改良

1. 何点の学生が何人いるかがわかるように頻度表を作る。
点数の範囲を $0 \sim 100$ としたとき、`juni[0]~juni[100]`の配列要素を初期化して、点数を配列の添字とし、頻度を配列の要素とする。
2. 何点以上の学生が何人いるかがわかるように累積頻度表を作る。
`juni[101]`の要素に順位1位を示す初期値1を入れておき、`juni[100]`から順番に`juni[0]`までの各要素に対して、頻度表の人数を加えていく。
3. 累積頻度表から順位を決定する。頻度表と累積頻度表を同じ配列で作成しているため、90点の学生の場合`juni[91]`を、66点の学生の場合`juni[67]`を参照する。

リスト2

```
#include <stdio.h>
#define Num    10
#define Max   100
#define Min    0

int main (void) {
    int a[] = {66, 56, 78, 55, 88, 90, 56, 76, 73, 58};
    int i, juni[Max+2];

    for (i = Min; i <= Max; i++)
        juni[i] = 0;    // 配列の初期化

    for (i = 0; i < Num; i++)
        ②;    // 頻度表の作成

    juni[Max+1] = 1;
    for (i = Max; i >= Min; i--)
        ③;    // 累積頻度表の作成

    printf(" 得点  順位¥n");
    for (i = 0; i < Num; i++)
        printf("%d  %d¥n", a[i], ④); // 順位の出力
    return 0;
}
```

リスト3

②		③		④	
---	--	---	--	---	--

3. 前問2のプログラムは正の整数で表される点数に対して、点数が高い順に順位付けするものであった。リスト4のプログラムは負の整数で表される点数も扱い、点数が低い順に順位付けするものである。点数の範囲を-20点~30点とする。

リスト4のプログラム中の⑤~⑦に該当する処理をページ下の解答欄に書きなさい。

```

#include <stdio.h>
#define Num    10
#define Max    30
#define Min    -20
#define Bias   1-(Min)           // 最小値を配列の1番目に対応させる

int main (void) {
    int a[] = {16, -6, 28, 5, -12, -10, 6, 26, 23, 8};
    int i, juni[Max+Bias+1];

    for (i = Min+Bias; i <= Max+Bias; i++)
        juni[i] = 0;           // 配列の初期化

    for (i = 0; i < Num; i++)
        ⑤;                   // 頻度表の作成

    juni[0] = 1;
    for (i = Min+Bias; i <= Max+Bias; i++)
        ⑥;                   // 累積頻度表の作成

    printf(" 得点  順位¥n");
    for (i = 0; i < Num; i++)
        printf("%d  %d¥n", a[i], ⑦); // 順位の出力
    return 0;
}
    
```

リスト4

⑤		⑥		⑦	
---	--	---	--	---	--

4. 以下は情報の流れる方向によってデータ伝送方式の形式を分類したものである。(1)~(3)の名称をそれぞれ答えよ。

- (1) トランシーバのように1本の伝送路で時間的に送信と受信を切り換えてデータの伝送を行う形式である。
- (2) 電話に代表され、同時に両方のデータの伝送を行う形式である。
- (3) テレビ放送などに代表され、一方向のみデータの伝送を行う形式である。

(1)	
(2)	
(3)	

5. 図1はLANの接続形態(ネットワークトポロジ)を抽象化した図とその特徴を示している。それぞれのネットワークトポロジの名称を答え、特徴を(a)~(c)より選択し、例にならって次のページの表を埋めよ。ただし、端末および交換機等をノードとして“●”,その伝送路をリンクとして“—”で示す。

《ネットワークの形態》

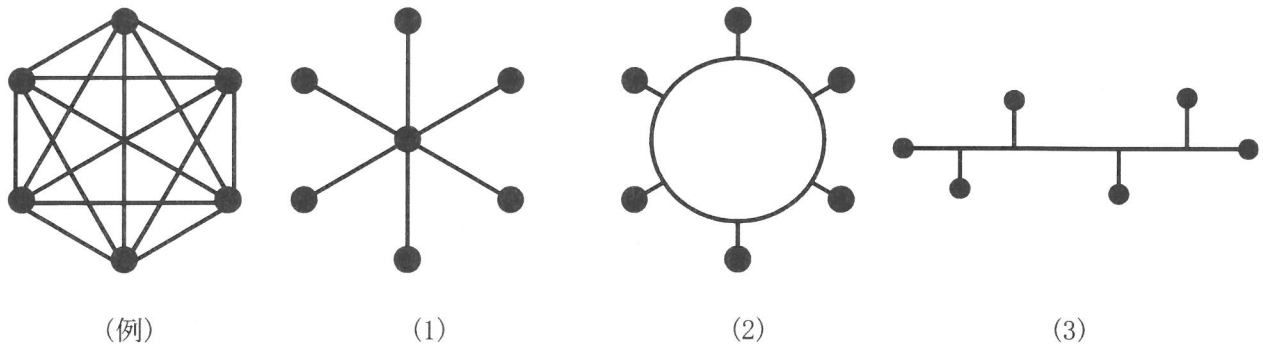


図1. ネットワークトポロジを抽象化した図

《特徴》

- (a) ハブなどの集線機器を中心に放射状にノードを接続する形態である。各端末から機器までをケーブルでつないでいるため、1本のケーブルに障害が発生してもネットワーク全体に問題が発生することはない。しかし、中心に設置された集線機器に障害や故障が発生するとネットワーク全体の機能がダウンする。
- (b) バスと呼ばれるケーブルに各ノードを接続する形態である。バスの終端は信号の反射を防ぐための終端抵抗(ターミネータ)が必要である。各ノードとバスをつなぐケーブルに障害が発生しても、ネットワーク全体には影響しない。しかしバスに問題が生じるとネットワーク全体の機能がダウンする。
- (c) バスがリング状になっており、各ノードはそのリングにつなぐ形態である。データは、一方向にリングを回っていき、接続されたノードは、データが自分宛てのものであれば、そのまま受け取るが、そうでないときは、またバスに送り返す。バスやバスにつながっているノードが故障するとネットワーク全体の機能がダウンする。

受験番号	
------	--

- (d) 各ノードを網の目状に接続する形態である。このように接続することによって各ノードが相互に通信する形態である。無線 LAN の通信機能を持ったノード同士で形成されることが多い。この構造は中心的なノードやリンクがないため通信経路の探索や伝送制御を適切に行うための技術が必要である。

ネットワークの形態	ネットワークトポロジの名称	特徴
(例)	メッシュ型	(d)
(1)		
(2)		
(3)		