98 年度 情報処理 理科一類 29, 30 組 試験問題 (100 点満点)

注意

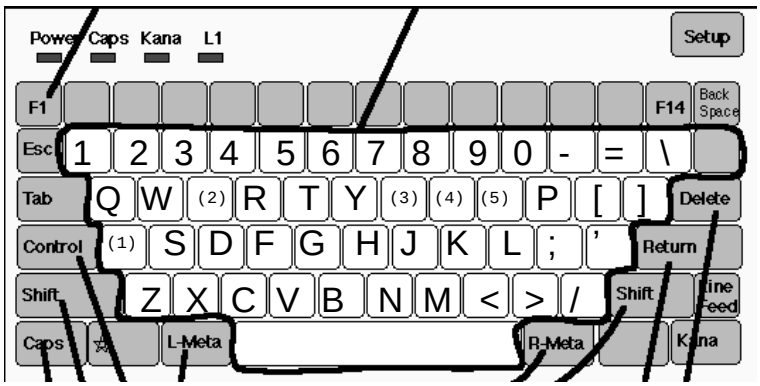
- 筆記用具以外の持ち込みは不可
- 解答用紙は 1 枚.

第 1 問 (15 点)

インターネットの代表的なアプリケーションとして，電子メール，ネットワークニュース，WWW(World Wide Web) があるが，それぞれ伝達性，速報性，秘匿性などの性質が異なっている．これら 3 つについて，郵便 (封書, 葉書)，電話，テレビ，本，新聞雑誌，その他の既存のメディアで最も性質が似ていると思われるものをそれぞれ 1 つとりあげ，それとの類似点，相違点を 2 ~ 3 個程度箇条書きにしなさい (最も似ているというのは主観が入るので，こちらの想定したものと異なっていても適切な類似点，相違点が指摘されていれば正解とする).

第 2 問 (5 点)

下図は，通常使われるキーボードの配列を示すが (余計な太線は無視すること)，母音 5 文字 (A,I,U,E,O) が抜けている． キーボード中の空白 (1)-(5) がそれぞれ，どの母音に対応するかを答えなさい．



第 3 問 (15 点)

以下の文の空白部分 (a-e) を埋めなさい.

Unix でハードディスク等に保存されるデータは, ファイル (file) と呼ばれるまとまりを単位として扱われる．多くのファイルを扱うので、いくつかのファイルを ディレクトリ (directory) と呼ばれる仮想的な「(机の) 引き出し」 にまとめることができるようになっている．

ディレクトリを作成するには [(a)] (make directory) というコマンドを用いる．コマンドラインから，

```
[ (a) ] sub1
```

と入力すると， sub1 というディレクトリができる．

ディレクトリの中に更にディレクトリを作って，階層化することもできる．上で作ったディレクトリ sub1 の下の階層に sub-child というディレクトリを作るには，コマンドラインから，

```
[ (a) ] [ (b) ]
```

と入力すると作ることができる．そのディレクトリの下に,Happy.txt というファイルを作るには，

```
echo 'I am happy.' > sub1/sub-child/Happy.txt
```

のようにする．ディレクトリは，親ディレクトリが 1 つで子ディレクトリを複数持つことができるという木 (Tree) 構造をしている．親ディレクトリを下にして絵を描いた時の一番下にあたるディレクトリが Unix システム内には一つだけ存在し，ルートディレクトリ (root directory) と呼ぶ．

参照の基準としているディレクトリを、 カレント・ディレクトリ (current directory) と呼ぶ．cd(change directory) というコマンドを使うと，カレント・ディレクトリを変えることができる．たとえば，

```
cd sub1
```

を実行すると，先ほど作った Happy.txt は，[(c)]という名前で参照できるようになり，

```
cat [ (c) ]
```

とすると，ファイルの内容を表示することができる．

ファイルやディレクトリを指定するには，カレント・ディレクトリ を基準にした相対パス (path) 指定と絶対パス指定の 2 種類の方法がある．たとえば，

```
cd /tmp
[ (a) ] sub1
[ (a) ] sub2
echo 'I am happy.' > sub1/happy.txt
cd sub2
```

としてディレクトリを 2 つ，ファイルを 1 つ作ったあとで，作ったファイルを表示するには，絶対パス指定で，

```
cat [ (d) ]
```

とすることもできるし，相対パスで

```
cat [ (e) ]
```

とすることもできる．

第 4 問 (10 点)

以下の文の空白部分 (a)-(e) を埋めなさい.

テキストファイルの形式は人間には扱い易いが, 計算機には扱いにくい. プログラム言語処理系は, 大きく分けると, プログラムの実行のたびにテキストファイルを読み込んで実行するインタプリタと, テキストファイルから機械語からなるファイルに変換する [(a)]がある. この時, プログラムを記述するテキストファイルのことをソースファイル (source file) と呼び, [(a)]によって作られる機械語のファイルのことをオブジェクトファイル (object file) と呼ぶ. [(a)]を使ってソースファイルからオブジェクトファイルを 作成することを「コンパイルする」と言う.

Java 言語では，Foo というクラス名のプログラムを作る時のソースファイルは，[(b)]という拡張子をつけて，Foo.[(b)]というファイル名にする必要がある．これを，javac という[(a)]でコンパイルすると，Foo.[(c)]という名前のオブジェクトファイルができる．このオブジェクトファイルは，Java 仮想機械の機械語のファイルになっている．

第 5 問 (10 点)

1 から 1000 までの素数を求めようと以下のプログラムを書いた．

```
public class Primes{
  public static void main(String[] argv){
   int n=1000,i,k;
   boolean p[]=new boolean[n];
   for(i=2; i<=n; i++) p[i]=true;
   for(i=2; i<=n; i++)
     if(p[i]){
        System.out.pritn(i+" ");
        for(k=i;k<=n;k=k+i) p[k]=true;
     }
  }
}
```

これをコンパイルしようとすると，

```
Primes.java:8: Method pritn(java.lang.String) not found in class java.io.PrintStream.
        System.out.pritn(i+" ");
                   ^
1 error
```

というエラーが出たので，プログラムを 1 行修正した．コンパイルに成功したので，実行させてみようとすると，

```
ktanaka@ecc-xs00> kaffe Primes
java.lang.ArrayIndexOutOfBoundsException
        at Primes.main(5)
```

というエラーが出た．そこで，プログラムを 1 行修正してコンパイルし直したところ正しく実行できた．
1 つめの修正と，2 つめの修正それぞれについて，修正後の行を書きなさい．

第 6 問 (10 点)

次のプログラムは後出しジャンケンのプログラムを配列を使って書き直したものである．正しく動くように空白 (a), (b) を埋めなさい．

```
import java.io.*;

public class Janken{
  static public void main(String args[]) throws IOException{
    DataInputStream istream=new DataInputStream(System.in);
    String hands[]={"Goo","Paa","Choki","Goo"};
    int i;
    System.out.print("Input (Goo, Choki, Paa) ");
    System.out.flush();
    String hand=istream.readLine();
    for(i=0;  (a)     ;i++){
      if(hand.equals(hands[i])){
        System.out.println("I selected "+  (b)      +". I win.");
        break;
      }
    }
    if(i==3){
```

```
      System.out.println("Your input was wrong.");
    }
  }
}
```

第 7 問 (15 点)

以下のラインエディタのプログラムは，d コマンド (カーソルの位置から指定された数の文字を削除 (Delete) する．指定した数が大き すぎる場合は，行末まで削除する) の処理の部分が抜けている．
抜けている部分を埋めて，プログラムを完成させなさい (答案用紙には抜けている部分だけを書くこと)．

```
import java.io.*;
public class LineEditor{
  static public void main(String[] args) throws IOException{
    DataInputStream istream=new DataInputStream(System.in);
    String str="";
    int cursor=0;
    while(true){
      System.out.println(str);
      int i;
      for(i=0;i<cursor;i++){
        System.out.print(" ");
      }
      System.out.println("^");
      System.out.println("Command: ");
      String command=istream.readLine();
      if(command.equals("q")){
        break;
      }
      else if(command.equals("i")){
        System.out.println("Insert what?");
        String subcommand=istream.readLine();
        int len=str.length();
        str=str.substring(0,cursor)+subcommand+str.substring(cursor,len);
      }
      else if(command.equals("d")){
      // d コマンドの処理
      }
      else if(command.equals("f")){
        System.out.println("Forward how many characters?");
        int num=Integer.parseInt(istream.readLine());
        int len=str.length();
        cursor+=num;
        if(cursor>len){
          cursor=len;
        }
      }
      else if(command.equals("s")){
        System.out.println("Search what?");
        String subcommand=istream.readLine();
        int newcursor=str.indexOf(subcommand,cursor);
        if(newcursor!=-1){
          cursor=newcursor;
        }
```

```
      }
    }
  }
}
```

第 8 問 (10 点)

以下のプログラムは，ある図形を表示するアプレットのプログラムである．表示される図形の概形を書きなさい (厳密に座標を書く必要はない)．

```
// <applet code=Q1 width=400 height=400></applet>
import java.awt.*;
import java.applet.*;
import java.lang.*;
public class Q1 extends Applet{
  public void paint(Graphics g){
    int i;
    for(i=0;i<5;i++){
      g.drawLine((int)(200-150*Math.sin((double)(i)*Math.PI*0.4)),
                 (int)(200-150*Math.cos((double)(i)*Math.PI*0.4)),
                 (int)(200-150*Math.sin((double)(i+3)*Math.PI*0.4)),
                 (int)(200-150*Math.cos((double)(i+3)*Math.PI*0.4)));
    }
  }
}
```

ヒント

- Math.sin は，ラジアンでの角度を引数として与えると sin を返す関数．
- Math.cos は，ラジアンでの角度を引数として与えると cos を返す関数．
- Math.PI は，円周率 (3.14159265358979323846) を表す定数．
- ‘‘(double) 式’’ で，整数値から実数値への変換をおこなう．
- ‘‘(int) 式’’ で，実数値から整数値への変換 (四捨五入) をおこなう．

第 9 問 (10 点)

授業の教材にも使ったニム (Nim) という名前の石取りゲームを取り上げる．ニムのゲームの規則は以下の通り

- 石が数列並んでいる
- プレイヤーは交代で同じ列の石を 1 から 3 個取る
- 最後の石を取ったプレイヤーの勝ち

このゲームは，始めの配置によって先手と後手のどちらが勝つかが決まっている．たとえば，列の数が 1 つのときには

- 石の数が 4n の時 → 後手の勝ち
- それ以外の時 → 先手の勝ち

となる．

(1) 列の数を 2 つにして，それぞれの列の長さを n, m とした時にどちらが勝つかを判断する方法を書きなさい．証明は必要ない．

(2) (1) に従ってプレイする Nim のプログラムを作るためのアルゴリズムを記述しなさい．記述方法は自然言語で書いてもいいし，プログラミング言語風に書いても構わない．