97 年度 情報処理 理科一類 16，29 組 試験問題 (110 点満点)

注意

- 筆記用具以外の持ち込みは不可
- 解答用紙，計算用紙は各 1 枚．

第 1 問 (20 点)

Unix システムでは， login 認証のためのパスワード s から一方向関数 f を用いて $f(s)$ を計算しそれをファイルに記録している．一方向関数とは，逆関数 f^{-1} を効率的に計算する方法がないような関数である．入力したパスワードの正当性は， f を適用した結果が記録したものと等しいかどうかでチェックする．

$f(s)$ が分かっても s を少しずつ変えながら $f(s)$ を計算し一致するかどうかをチェックしないと s を求めることはできないため，古い Unix システムでは，暗号化されたパスワード $f(s)$ を特に隠さず，一般ユーザに見えるファイルに置かれていた．これが，最近の計算機の速度の向上によって，危険になってきたことを確かめる．

暗号化されたパスワードが得られて，元々のパスワード s が以下の条件を満たすことがわかった時，関数 f を 1 秒間に 100 万回実行できるコンピュータで，元のパスワードを求めるには平均何秒かかるか? 有効数字 1 桁で答え，計算過程も示しなさい．

(a) 英小文字 (a,b,...,z) だけからなる 6 文字以下のパスワード

(b) 英小文字 (a,b,...,z)，英大文字 (A,B,..,Z)，数字 (0,1,..,9) だけからなる 8 文字以下のパスワード

(c) 4 文字以下の英単語 (仮に 4000 個とする) を 2 つつなげたパスワード

(d) 4 文字以下の仮名文字 (「が」，「ょ」などは 1 文字と数える) を訓令式ローマ字であらわしたパスワード

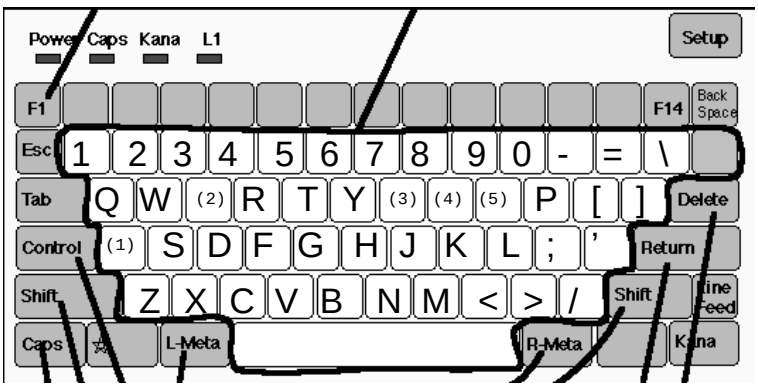
第 2 問 (10 点)

WWW ブラウザで WWW ドキュメントを閲覧することを指して「インターネットする」という言葉が使われることがあるが，この表現は適当でないと主張する人もいる. 次のキーワードをすべて使って, 200 文字程度でこの主張を代弁しなさい (個人的に反対意見を持っていても，自分の意見を書くのではなく，相手がどう主張するかを予想して書きなさい).

キーワード： インターネット, WWW, メール, ニュース, 電話, FAX, 電話線

第 3 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 4 問 (10 点)

Unix の wc というコマンドの使い方を知るために， kterm のウィンドウで， man wc を実行したところ，説明が英語で出力された．その一部を以下に示す．これを日本語に翻訳しなさい．

> wc counts the number of bytes, whitespace-separated words, and newlines in each given file, or the standard input if none are given or when a file named '-' is given. It prints one line of counts for each file, and if the file was given as an argument, it prints the filename following the counts.

注)

byte バイト (2 進数 8 桁の数，ファイル操作の最小単位はバイトとなる)

whitespace 空白文字 (スペース，タブ，改行など)

newline 改行文字

standard input 標準入力 (コマンドが標準的に受け付ける入力．通常はキーボードからの入力がこれに対応するが，コマンド行で '<' や '|' を使うとファイルや他のコマンドの出力に置き換えることができる)

argument 引数 (Unix のコマンドではコマンド名の後に空白で区切って入力する．「ひきすう」と読む)

第 5 問 (10 点)

コンピュータの構成要素 (1)-(4) に対応する説明を (a)-(d) の中から選びなさい

(1) CPU (Central Processing Unit), 中央処理装置

(2) メモリ (Memory) 記憶装置

(3) I/O (Input/Output Device) 入出力装置

(4) バス (Bus)

(a) (c) と (b) や (d) との間でやりとりするデータが流れるところ.

(b) (c) が読み込むプログラム や, プログラムの実行に用いるデータを格納する.

(c) (b) から機械語と呼ばれる 2 進数のプログラムを読み込んで, その 指示に従って, (b) や (d) とデータのやりとりをおこなう.

(d) キーボード, マウス, ディスプレイ, サウンドボードなど人間 との対話 (interaction) のための装置の他に, コンピュータ同士の接続のための ネットワーク装置や, 外部記憶装置も含まれる

第 6 問 (10 点)

次のプログラムを実行した結果が，(a)-(e) のうちのどれになるかを答えなさい．

```
public class Sample {
  public static void main(String[] argv){
    int x;
    x=2;
    System.out.println(x+1);
    x=x+2;
    System.out.println(x+1);
  }
}
```

(a) 3 / 3　　(b) 3 / 5　　(c) x+1 / x+1　　(d) 5 / 5　　(e) 2 / 4

第 7 問 (10 点)

上の「図形」を表示する以下のプログラム (Test6) は 8 行めが抜けている．

```
public class Test6{
  public static void main(String[] argv){
    int x,y;
    for(y=0;y<6;y=y+1){
      for(x=0;x<y;x++)
       System.out.print(" ");
      System.out.print("*");
//（この行が抜けています．）
      System.out.println("*");
    }
  }
}
```

抜けている行を下から選びなさい．

(a) `for(x=0;x<6;x++){System.out.print(" ");}`

(b) `for(x=0;x<y;x++){System.out.print(" ");}`

(c) `for(x=y+1;x<11-y;x++){System.out.print(" ");}`

第 8 問 (15 点)

以下は 1000 までの素数を求めるプログラム (エラトステネスのふるい) である．

```
public class Test9{
  public static void main(String[] argv){
   int n=1000,i,k;
   boolean p[]=new boolean[n+1];
   for(i=2; i<=n; i++) p[i]=true;
   for(i=2; i<=n; i++)
     if(p[i]){
        System.out.print(i+" ");
//（この部分が抜けています）
     }
  }
}
```

抜けている部分にあたるプログラムを書きなさい．1 行で書く必要はありません．

ヒント) p[i] の要素が true の時は今のところ合成数とは判明していないことを，false の時は合成数だということを表すようにプログラミングされています．抜けている部分では，n までの i の倍数を合成数だと指定します．

第 9 問 (20 点)

あるメソッドの中で, 自分のメソッドを呼び出すことを「再帰 (recursion)」と呼び, プログラムの実用上のテクニックとしても, プログラムの理論的解析においても重要な概念である. 以下は再帰を使ってある関数を計算するプログラムである.

```
public class Test12{
  static int f(int n){
    if(n<2) return 1;
    else return f(n-1)+f(n-2);
  }
  public static void main(String[] argv){
    if(argv.length >=1){
      System.out.println(f(Integer.parseInt(argv[0])));
```

```
    }
  }
}
```

(1) このプログラムをコンパイルして，

```
kaffe Test12 5
```

と引数 5 を与えて実行した時に表示される数を計算過程も含めて示しなさい．計算過程の表記法は任意とする

(2) 同じ関数 f を再帰を用いずに書いたのが，以下のプログラムである．

```
  static int f(int n){
   if(n<2) return 1;
   int fv[]=new int[n+1];
   fv[0]=1; fv[1]=1;
   int i;
   for(i=2;i<=n;i++){
// この行が抜けています．
   }
   return fv[n];
  }
```

抜けている行を埋めなさい．

(3) (2) は (1) よりも効率の良いプログラムになっている．このことを，'+' の演算の回数を用いて説明しなさい．