

学籍番号 _____ 氏名 _____

1 UNIX コマンドとコンパイル

(1) 以下の文章の () に適当な語句を埋めよ。(各 2 点)

- カレントディレクトリーのパスを調べるコマンドは、(`pwd`) です。
- 親ディレクトリーへ移動するコマンドは、(`cd ..`) です。
- サブディレクトリー hoge hoge に移動するコマンドは、(`cd hoge hoge`) です。
- カレントディレクトリーにあるファイルやサブディレクトリーの名前を調べるコマンドは、(`ls`) です。
- ファイルを削除するコマンドは、(`rm hoge hoe`) です。これで、hoge hoge というファイルが削除されます。

(2) C 言語のソースファイル「hoge.c」をコンパイルして、「calc」という実行ファイルを作りたい。以下の場合に応じて、そのコマンドを記述しなさい(各 2 点)

- hoge.c に数学関数が含まれていない場合
`cc -o calc hoge.c`
- hoge.c に数学関数が含まれている (math.h が有る) 場合
`cc -lm -o calc hoge.c`

2. C 言語

(1) 基礎 (各 3 点)

- 識別子とは、変数や記号定数、関数などにつける名前です。この名前に用いることができる文字を示しなさい。
英大文字 A~Z 英小文字 a~z
数字 0~9 アンダースコア _
- コメント文を書く方法を示せ。
文を /* と */ で囲む

(2) 変数とデータの型 (6 点, 8 点)

• 下表の変数宣言の表を埋めよ。

| 型名 | 変数名 | 変数宣言 |
|--------|------|---------------------------|
| 文字型 | a, b | <code>char a, b;</code> |
| 整数型 | i, j | <code>int i, j;</code> |
| 倍精度実数型 | x, y | <code>double x, y;</code> |

• 変数の適用範囲に関して、①~⑧を選択して表を埋めよ。

| 種類 | 宣言場所 | メモリ配置 タイミング | メモリ消滅 タイミング | 変数適用 範囲 |
|----|------|----------------|----------------|------------|
| ② | 関数内 | ③ | ⑤ | ⑧ |
| ① | 関数外 | ④ | ⑥ | ⑦ |

- ①外部変数 ②自動変数
- ③関数コール時 ④プログラム起動時
- ⑤関数処理終了時 ⑥プログラム終了時
- ⑦どの関数からでも使用可能 ⑧宣言した関数のみ

(3) 制御文 (各 2 点)

- for 文の実行順序で正しいのはどれか?
`for (初期値; 継続条件式; 再設定式) {文}`

- ① 初期値 → 文 → 再設定式 → 継続条件判断
- ② 初期値 → 文 → 継続条件判断 → 再設定式
- ③ 初期値 → 継続条件判断 → 再設定式 → 文
- ④ 初期値 → 継続条件判断 → 文 → 再設定式

答え ④

- do while 文で正しいのはどちらか。
`do (条件式) while {文}`

- ① 最初は、文を実行後、条件式を判断する。
- ② 最初は、条件式判断後、文を実行する。

答え 出題ミス全員に加点

(4) 配列 (6 点, 2 点)

- 下表の配列宣言の表を埋めよ。ただし、配列の型は倍精度実数型とし、配列名は x とする。

| 次元 | 要素数 | 配列宣言 |
|----|-------------|---------------------------------------|
| 1 | 100 | <code>double x[100];</code> |
| 2 | 100×100 | <code>double x[100][100];</code> |
| 3 | 100×100×100 | <code>double x[100][100][100];</code> |

- 宣言により要素数 100 の配列を用意した。配列の添え字として使える番号の範囲を示せ。

0~99 までが使用できる

(5) 関数 (3 点, 3 点, 4 点, 2 点)

- C 言語のプログラムに必ず書かれる関数名を示しなさい。
`main`

- 以下の条件で、関数を作成する。プロトタイプ宣言を書け。

- 関数の名前は、myfunc
- 戻り値は、倍精度実数
- 引数は、倍精度実数が 1 個と、整数が 2 個。変数名は、適当でよい(受験者が決めろ)。

`double myfunc(double x, int i, int j);`

- 関数の呼び出す側の実引数と、呼ばれる側の仮引数の書き方を表に示せ。実引数を a, 仮引数を y とする。仮引数の型は不要である。

| データの渡し方 | 実引数 | 仮引数 |
|---------|-----|-----|
| 値渡し | a | y |
| アドレス渡し | &a | *y |

- データの渡し方のうち、実引数に影響を与えるのは、どちらか(値渡し or アドレス渡し)?

アドレス渡し

3. C 言語のプログラム

- (1) ディスプレイに「Hello World !!」と書き出すプログラムを作成しなさい。かぎ括弧は書き出し不要 (15 点)
<注意> 全て分からなくても、分かる範囲をできるだけ、記述すること。

```
#include <stdio.h>
main()
{
    printf("Hello World!!");
}
```

- (2) 1~100 の和を求めて、ディスプレイに書き出すプログラムです。下線ア、イを示せ。(各 2 点)

```
#include <stdio.h>
main()
{
    int a, b;

    a = b = 0;

    ア (a=1; イ; a++){
        ウ;
    }

    printf("b = %d\n",b);
}
```

答 (ア) for
(イ) a <= 100
(ウ) b += a

- (3) (2) のアと異なる命令を用いて、1~100 の和を求め、その結果を表示するプログラムを示せ。ここは、努力の割に配点は低いので、他の問題ができた後、この問題を解くこと。(5 点) 以下 2 つの例を示す。

```
#include <stdio.h>
main()
{
    int a, b;

    a = 1;
    b = 0;

    while(a<=100){
        b += a;
        a++;
    }

    printf("b = %d\n",b);
}

#include <stdio.h>
main()
{
    int a, b;

    a = b = 0;

    next: b+=a;

    if(a<100){
        a++;
        goto next;
    }

    printf("b = %d\n",b);
}
```

- (4) アドレスを渡すことにより、main 関数の変数 a と b の値を関数 swap で交換している。引数である下線アとイを示せ。更に交換に必要な文下線ウを示せ。(各 2 点)

```
#include <stdio.h>
void swap(イ);
main()
{
    int a, b;

    a=1;
    b=-1;

    swap(ア);
    printf(" %d %d\n",a,b);
}
```

```
void swap(イ)
{
    int c;

    ウ;
    *a=*b;
    *b=c;
}
```

答 (ア) &a,&b
(イ) int *a, int *b
(ウ) c=*a;

- (5) ファイル(calresult)に、変数の値と三角関数の値を書き出している。ファイル処理に関する下線ア~オを示せ。(各 2 点)

```
#include <stdio.h>
#include <math.h>
main()
{
    ア *out;
    double pi=4*atan(1);
    double theta, s, c, t;
    int i;

    out = イ ("ウ", "w");

    for(i=0;i<=100;i++){
        theta = i*pi/100;
        s = sin(theta);
        c = cos(theta);
        t = tan(theta);
        エ (out, "%f¥t%f¥t%f¥t%f¥n",theta,s,c,t);
    }

    オ (out);
}
```

答 (ア) FILE
(イ) fopen
(ウ) calresult
(エ) fprintf
(オ) fclose