

学籍番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

1. UNIX の使い方 (計 30 点)

1.1 ファイル(計 20 点)

[問 1] (2 点)

`/D1/D4/f1`

[問 2] (2 点)

`../../D1/D4/f1`

[問 3] (2 点)

`/D2/D6`

[問 4] (2 点)

`D2`

[問 5] (2 点)

`D9 と D10`

[問 6] (2 点)

`cd ..`

[問 7] (2 点)

以下のようにすれば、サブディレクトリーに移れる。

`cd サブディレクトリー名`

[例]D9 というサブディレクトリーに移る

`cd D9`

[問 8] (2 点)

`cd`

[問 9] (4 点)

ファイルとはプログラムやデータのように、コンピューターで取り扱う実体である。それに対して、ディレクトリーは、ファイルの入れ物である。ディレクトリーにはファイルのほか、ディレクトリーも格納でき、階層構造が可能となっている。

1.2 コマンド(計 10 点)

[問 1] (2 点)

`ls`

[問 2] (2 点)

`mkdir hoge`

[問 3] (2 点)

`cp hoge.c fuga.c`

[問 4] (2 点)

`rm hoge.c`

[問 5] (2 点)

`mv hoge.c fuga.c`

2. C 言語の文法 (計 45 点)

2.1 基礎(計 12 点)

[問 1] (4 点)

配列とは同じ型のデータが複数ある場合に使われるデータ構造である。配列名と自然数の添え字により、データにアクセスできる。

[問 2] (4 点)

コメント文を `/*` と `*/` で囲む。

`/* ここにコメントを書く */`

[問 3] (2 点)

`int i;  
double x;`

[問 4] (2 点)

`double hoge[1001];`

## 2.2 制御文 (計 15 点)

[問 1] (5 点)

```
if(a > max){
    max = a;
}
```

[問 2] (5 点)

```
if( 50<=a && a<=100){
    a=75;
}
```

[問 3] (5 点)

```
if(a <= -50 || 50 <=a){
    a=999;
}
```

## 2.3 関数 (計 18 点)

[問 1] (6 点)

```
k=13    m=71
```

[問 2] (各 2 点)

[ア] 1

[イ] 2

[ウ] 4

[エ] 3

[オ] 6

[カ] 5

## 3. プログラム作成 (計 25 点)

### 3.1 基本 (15 点)

```
#include <stdio.h>

int main(void){
    int sum, i, finish;

    finish = 1000;

    sum=0;          /* 初期化 */

    for(i=1; i<=finish; i++){
        sum+=i;
    }

    printf("1～ %d までの和は、%d です。 \n", finish, sum);

    return 0;
}
```

### 3.2 素数(10点)

```
#include <stdio.h>
#define TEST 1000001

int main(void){
    int number[TEST];
    int i, j;

    /*---- 初期化 ----*/
    for(i=1; i<TEST; i++){
        number[i]=0;
    }

    /*---- 素数判定 ----*/
    for(i=2; i<TEST; i++){
        if(number[i]==0){
            for(j=2*i; j<TEST; j+=i){
                number[j]=1;
            }
        }
    }

    /*---- 結果表示 ----*/
    j=0;
    printf("No\tPrime Number\n");

    for(i=2; i<TEST; i++){
        if(number[i]==0){
            j++;
            printf("%d\t%d\n", j, i);
        }
    }

    return 0;
}
```