

学籍番号 _____ 氏名 _____

1. 制御文 (各 5 点)

[問 1]

```
if(a < b){
  a=1;
  b=-1;
}
```

[問 2]

```
if(a < b){
  a = 1;
}else if(a == b){
  a = 0;
}else{
  a = -1;
}
```

2. 配列

[問 1] (5 点)

単純型の場合、格納できるデータは 1 個のみで、変数名を指定することによりそのデータにアクセスする。一方、配列の場合、同じ型のデータが複数個格納でき、配列名と自然数の添え字を指定することによりデータにアクセスする。

[問 2] (5 点)

大量のデータを扱う場合、単純型に比べ配列の方が以下の点で有利である。

- 変数の宣言も簡単である。
- 配列名と自然数によりデータにアクセスできるので、大量のデータの操作が簡単である。

[問 3] (各 5 点)

hoge[0] ~ hoge[100] の場合

```
double hoge[101];
```

fuga[0][0] ~ fuga[100][100] の場合

```
int fuga[101][101];
```

3. 繰り返し (各 10 点)

[問 1]

```
#include <stdio.h>
int main(void){
  int i, sum;

  sum = 0;

  for(i=1; i<=10000; i++){
    sum += i;
  }

  printf("sum = %d\n", sum);

  return 0;
}
```

[問 2]

```
#include <stdio.h>
int main(void){
  int i, sum;

  sum = 0;
  i = 1;

  while(i<=10000){
    sum += i;
    i++;
  }

  printf("sum = %d\n", sum);

  return 0;
}
```

4. プログラム作成

4.1 温度のデータ処理 (各 2 点)

[ア] 4	[イ] 1	[ウ] 3	[エ] 1	[オ] 4
[カ] 3	[キ] 2	[ク] 4	[ケ] 1	[コ] 4
[サ] 3	[シ] 2	[ス] 2	[セ] 4	[ソ] 1

4.2 アクセスカウンター (20 点)

```
#include <stdio.h>

int main(void){
    FILE *fp;
    int access;

    /*----- ファイルのオープン(読み込みモード) -----*/

    if((fp=fopen("/tmp/1e/up.log","r"))==NULL){
        printf("can not open the file\n");
        return 1;
    }

    fscanf(fp, "%d", &access);    /*-- ファイルからデータを読み込む --*/

    fclose(fp);                  /* -- ファイルのクローズ --*/

    /*----- 実行回数の表示 -----*/

    printf("%d 回目の実行です\n", access);

    /*----- ファイルのオープン(書き込みモード) -----*/

    if((fp=fopen("/tmp/1e/up.log","w"))==NULL){
        printf("can not open the file\n");
        return 1;
    }

    fprintf(fp, "%d",access+1);    /*-- ファイルへデータを書き込む --*/

    fclose(fp);                  /*-- ファイルのクローズ --*/

    return 0;
}
```