第10回
プログラミング及び実習 II

第9回小テストの結果

小テストに関するコメント

コンパイルできない不完全なプログラムなど
→ プログラミングに慣れていない、あるいは、複雑な問題に対しては、
要件をバラして段階的に
　　プログラムを作成→コンパイル→実行→結果を確認
することを繰り返してプログラムを完成させる。

3つの実数 x, y, z の値を順に入力すると、(x+y)z と √(x^2+..)の値を出力することを繰り返すプログラム...。ただし、x, y, z の値にすべて 0 が入力されたときはプログラムを終了するものとする。

1. x,y,z を入力して x,y,z の値を出力するプログラム
2. x,y,z を入力して (x+y)z の値を出力するプログラム
3. 3 に加えて、x^2+y^2+z^2 の値を出力するプログラム
4. 3 に加えて、√(x^2+y^2+z^2) の値を出力するプログラム
5. 1〜4を繰り返すプログラム
6. 5 に加えて、x, y, z 全てが 0 のとき終了するプログラム

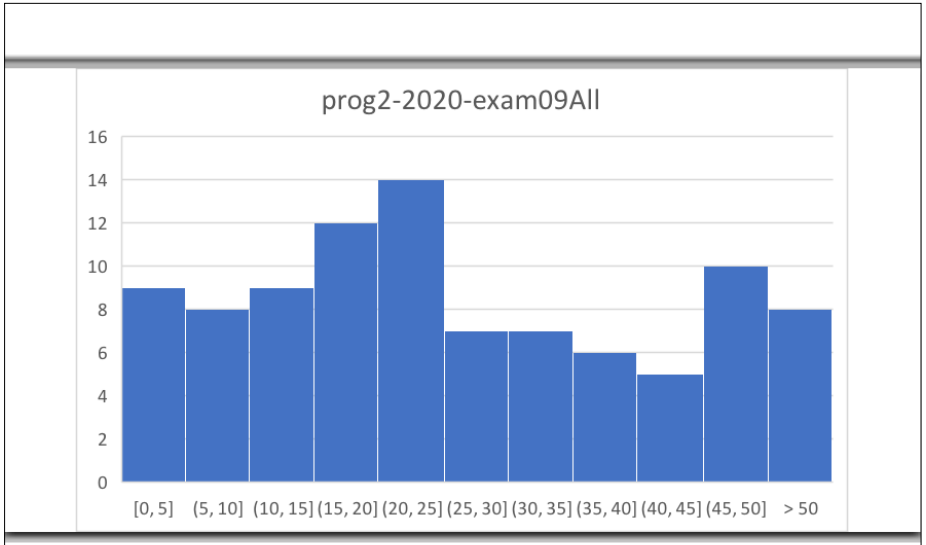
小テストに関するコメント

プログラミングが書けない人ほどインデント(字下げ)しない
→ プログラムコードは人間のためのものです

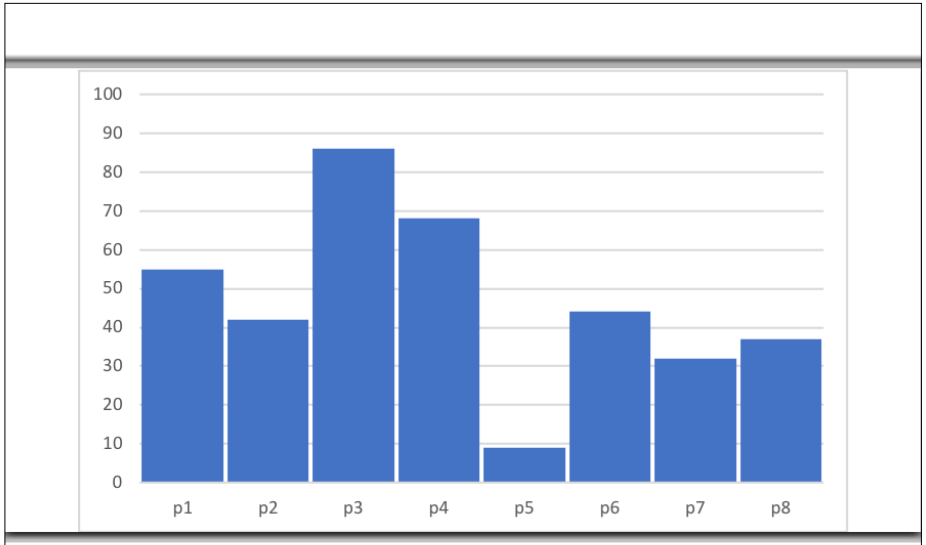
```
#include <stdio.h>

int main() {
  int a;
  scanf("%d", &a);

  return a/2;
}
```

```
#include<stdio.h> int main(){int a;scanf("%d",&a);return a/2;}
```

```
0016140 0000 0000 0000 0000 0099 0000 0001 0012
0016160 0e10 0060 0000 0000 0000 0000 0000 0000
0016200 00b8 0000 0004 fff1 0000 0000 0000 0000
0016220 0000 0000 0000 0000 0001 0000 0004 fff1
0016240 0000 0000 0000 0000 0000 0000 0000 0000
0016260 00c3 0000 0001 0011 0740 0040 0000 0000
0016300 0000 0000 0000 0000 00d1 0000 0001 0014
0016320 0e20 0060 0000 0000 0000 0000 0000 0000
0016340 0000 0000 0004 fff1 0000 0000 0000 0000
0016360 0000 0000 0000 0000 00dd 0000 0000 0012
                        :
```



評価の分布



回答数の分布（>0）

exam09 のフォローアップ提出

Sharif-Judge での exam09 の自己フォローアップ提出を受け付けます（小テスト終了時間から30日）

得点率は小テスト実施時（各10点満点）の25%

exam09 も最終提出で評価（提出間違いに注意！）

自己フォローを目的としてるため、コピーコードの提出があれば exam09 は0点評価（含む口頭試問）

第9回小テストの解答例

exam09-1：整数列の文字変換

```
#include <stdio.h>
#define N 100

int main() {
    int n;
    int a[N];

    for (n=0; n<N; n++) {
        scanf("%d", &a[n]);
        if (a[n] <= 0) break;
    }

    for (int i=0; i<n; i++) {
        if (a[i] < 'A' || 'Z' < a[i]) continue;
        else printf("%c", a[i]);
    }
    printf("\n");
    return 0;
}
```


exam09-2：数字の反転

```
#include <stdio.h>

int main() {
    int n;

    while (1) {
        scanf("%d", &n);
        if (n < 10) return 1;

        while (n > 0) {
            printf("%d", n%10);
            n = n/10;
        }
        printf("\n");
    }
    return 0;
}
```


exam09-3：総和でアスタリスク

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int sum = 0;
    for (int i=1; i<=n; i++) {
        sum += i;
        for (int j=0; j<sum; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```


exam09-4：奇数・偶数ごとの平均値

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    if (n <= 0) return 1;

    int val, num_o, num_e;
    double sum_o, sum_e;
    num_o = num_e = sum_o = sum_e = 0;
    for (int i=0; i<n; i++) {
        scanf("%d", &val);
        if (val%2) {
            sum_o += val;
            num_o++;
        } else {
            sum_e += val;
            num_e++;
        }
    }
    if (num_o)
        printf("%.3f\n", sum_o/num_o);
    else
        printf("\n");

    if (num_e)
        printf("%.3f\n", sum_e/num_e);
    else
        printf("\n");

    return 0;
}
```

exam09-5：kナッチ数列

```
#include <stdio.h>
#define ULIMIT 1000

int main() {
    int k;
    scanf("%d", &k);
    if (k < 2) return 1;

    int a[k];
    for (int i=0; i<k-1; i++) {
        a[i] = 0;
        printf("%d ", a[i]);
    }
    a[k-1] = 1;
    printf("%d ", a[k-1]);

    int i = k-1;
    while (1) {
        int sum = 0;
        for (int j=0; j<k; j++)
            sum += a[j];
        if (sum > ULIMIT) break;
        printf("%d ", sum);

        i++;
        a[i%k] = sum;
    }
    printf("\n");

    return 0;
}
```


exam09-6：円周率

```
#include <stdio.h>
#include <math.h>

int main() {
    int m;
    scanf("%d", &m);
    if (m < 0) return 1;

    int pm = -1;
    double sum = 0.0;
    for (int k=0; k<=m; k++) {
        pm *= -1;
        sum += pm / (2.0*k + 1);
        printf("%.20f\n", 4*sum);
    }
    return 0;
}
```


exam09-7：配列の並び替えと重複削除

```
#include <stdio.h>

#define N 100

int main() {
    int n, cnt[N];
    scanf("%d", &n);
    if (n < 1) return 1;

    for (int i=0; i<N; i++)
        cnt[i] = 0;

    int a[n];
    for (int i=0; i<n; i++) {
        do {
            scanf("%d", &a[i]);
        } while (a[i] < 0 || N <= a[i]);
        cnt[a[i]]++;
    }

    for (int i=0; i<N; i++)
        if (cnt[i]) printf("%d ", i);
    printf("\n");

    return 0;
}
```


exam09-8：2科目合計点の数え上げ

```
#include <stdio.h>
#define MAX 200
#define SKIP 20
#define N ((MAX/SKIP)+1)

int main() {
    int n;
    scanf("%d", &n);
    if (n <= 0) return 1;

    int dist[N];
    for (int i=0; i<N; i++)
        dist[i] = 0;

    int a[n], total[n];
    for (int i=0; i<n; i++)
        scanf("%d", &total[i]);

    for (int i=0; i<n; i++) {
        scanf("%d", &a[i]);
        total[i] += a[i];
        dist[total[i]/SKIP]++;
    }

    for (int i=0; i<N; i++)
        printf("%d\n", dist[i]);

    return 0;
}
```

第10回の内容

C言語の関数

関数を定義する

関数

数学の関数

プログラミング言語の関数　`int func(a)`

`int a`　　`func()`　　`int v`

プログラミング言語の関数

プログラミング言語の関数　`int a`　`func()`　`int v`

入力がない場合もある　`func()`　`int v`

出力がない場合もある　`int a`　`func()`

`scanf()`

別の入力や出力をする場合もある　`int a`　`func()`　`int v`

`printf()`

main() 関数

C言語で最初に呼び出される関数

実行可能なCプログラムを書く ＝ main() 関数を定義する

return 文によって関数の値（返り値）を定義する

main という(何もしない)名前の関数

```
int main() {
    return 0;
}
```

`main()`　`0`

main は int 型の0を（返り値として）返す

main は int 型を（返り値として）返す

第10回の内容

C言語の関数

簡単な関数を定義する

関数を定義する

これまでは、C言語に用意されていた標準関数を「呼び出して」いた
printf(), scanf(), exit(), sin(), sqrt(), ...

関数を自分で定義する（ソースコードとして書く）ことで、その新しい関数を呼び出して「何度でも」利用することができる

main() 関数：C言語で実行時に最初に「自動的に」呼び出される関数

実行可能なCプログラムを書く ＝ main() 関数を定義する

これまでは、ただ1つの関数 main() だけが定義されたソースコード

```
int main() {
     :
   return 0;
}
```

main()　0

関数の宣言と定義

```
#include <stdio.h>

void func();

int main() {
  func();
  return 0;
}

void func() {
  printf("Hello\n");
}
```

void 型の関数 func() のプロトタイプ宣言

関数 func() の呼び出し（使う）

関数 func() の定義

プロトタイプ宣言：
関数の名前や返り値の型(あるいは引数)などを宣言（他の関数などで利用できるように教える）する

main()　0

関数定義：
実際にその関数がどんな仕事をするのかをプログラムコードとして記述する

func()

関数と返り値とreturn文

C言語の関数は「型名 関数名() {...}」で定義される (ex. int func() {...})

C言語の関数は「関数名()」で呼び出される(利用される)

C言語の関数は（型指定された）値をもつ（もたないこともある → void型の関数）

return 文によって関数の値（返り値）が決まる → 返り値の型 = 関数の型名

scanf() 関数と printf() 関数も int 型の返り値をもつ

```
int a, n, m;

n = scanf("%d", &a);
m = printf("a is %d\n", a);
// n? m?
```

返り値のある関数の宣言と定義

```
#include <stdio.h>

int ten();

int main() {
  printf("%d\n", ten());
  return 0;
}

int ten() {
  return 10;
}
```

int 型の関数 ten() のプロトタイプ宣言

関数 ten() の呼び出しと返り値の利用

関数 ten() の定義

関数を評価する（呼び出す）と定義された型の値を（返り値として）もつ

返り値の値は、関数定義の return 文で与えられる

関数の型と return 文で返す式（値）の型は一致

main()　0

10

ten()