

第14回

プログラミング及び実習 II

1

第13回課題の解答例

2

ex13-1 : 3x3の掛け算表

```
#include <stdio.h>
#define N 3

void showtable(int a[N][N]);

int main() {
    int a[N][N];
    for (int i=0; i<N; i++)
        for (int j=0; j<N; j++)
            a[i][j] = 0;

    int n, m;
    while (1) {
        scanf("%d%d", &n, &m);
        if (n <= 0 || m <= 0 || n > N || m > N)
            break;
        a[n-1][m-1] = n*m;
        showtable(a);
    }
    return 0;
}

void showtable(int a[N][N]) {
    for (int i=0; i<N; i++) {
        for (int j=0; j<N; j++)
            printf("%2d", a[i][j]);
        printf("\n");
    }
    return;
}
```

3

ex13-2 : 行列の和

```
#include <stdio.h>
#define N 2

void mscan(int a[N][N]);

int main() {
    int a[N][N], b[N][N], c[N][N];
    mscan(a); mscan(b);

    for (int i=0; i<N; i++)
        for (int j=0; j<N; j++)
            c[i][j] = a[i][j] + b[i][j];

    for (int i=0; i<N; i++) {
        for (int j=0; j<N; j++)
            printf("%3d", c[i][j]);
        printf("\n");
    }
    return 0;
}

void mscan(int a[N][N]) {
    for (int i=0; i<N; i++)
        for (int j=0; j<N; j++)
            scanf("%d", &a[i][j]);
    return;
}
```

4

ex13-3 : 3x3のマス目

```
#include <stdio.h>
#define N 3

void showtable(int a[N][N]);

int main() {
    int a[N][N];
    for (int i=0; i<N; i++)
        for (int j=0; j<N; j++)
            a[i][j] = 0;

    int player = 1;
    int n, m;
    while (1) {
        scanf("%d%d", &n, &m);
        if (n <= 0 || m <= 0 || n > N || m > N)
            break;
        if (a[n-1][m-1] == 0)
            a[n-1][m-1] = player;
        else
            continue;

        showtable(a);
        player *= -1;
    }
    return 0;
}

void showtable(int a[N][N]) {
    for (int i=0; i<N; i++) {
        for (int j=0; j<N; j++) {
            if (a[i][j] == 1)
                printf("x");
            else if (a[i][j] == -1)
                printf("+");
            else
                printf(" ");
        }
        printf("\n");
    }
    return;
}
```

5

ex13-4 : 行列の和2

```
#include <stdio.h>

void mscan(int n, int a[n][n]);

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

    int a[n][n], b[n][n], c[n][n];
    mscan(n, a); mscan(n, b);

    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            c[i][j] = a[i][j] + b[i][j];

    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++)
            printf("%3d", c[i][j]);
        printf("\n");
    }
    return 0;
}

void mscan(int n, int a[n][n]) {
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            scanf("%d", &a[i][j]);
    return;
}
```

6

ex13-5 : 三目並べ

```
#include <stdio.h>
#include <stdlib.h>
#define N 3

void showtable(int a[N][N]);
int wincheck(int a[N][N]);

int main() {
    int a[N][N];
    for (int i=0; i<N; i++)
        for (int j=0; j<N; j++)
            a[i][j] = 0;

    int player = 1;
    int n, m;
    while (1) {
        scanf("%d%d", &n, &m);
        if (n <= 0 || m <= 0 || n > N || m > N)
            break;
        if (a[n-1][m-1] == 0)
            a[n-1][m-1] = player;
        else
            continue;

        showtable(a);
        if (wincheck(a)) break;

        player *= -1;
    }
    return 0;
}

void showtable(int a[N][N]) {
    for (int i=0; i<N; i++) {
        for (int j=0; j<N; j++) {
            if (a[i][j] == 1) printf("x");
            else if (a[i][j] == -1) printf("+");
            else printf(" ");
        }
        printf("\n");
    }
    return;
}

int wincheck(int a[N][N]) {
    int cross1 = 0;
    int cross2 = 0;
    for (int i=0; i<N; i++) {
        cross1 += a[i][i];
        cross2 += a[i][N-1-i];
        if (abs(cross1) == N || abs(cross2) == N)
            return 1;
        int tate = 0;
        int yoko = 0;
        for (int j=0; j<N; j++) {
            yoko += a[i][j];
            tate += a[j][i];
        }
        if (abs(yoko) == N || abs(tate) == N)
            return 1;
    }
    return 0;
}
```

7

ex13-6 : 行列の積

```
#include <stdio.h>

void mscan(int n, int a[n][n]);

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

    int a[n][n], b[n][n], c[n][n];
    mscan(n, a); mscan(n, b);

    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            c[i][j] = 0;
            for (int k=0; k<n; k++)
                c[i][j] += a[i][k]*b[k][j];
        }
    }

    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++)
            printf("%3d", c[i][j]);
        printf("\n");
    }
    return 0;
}

void mscan(int n, int a[n][n]) {
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            scanf("%d", &a[i][j]);
    return;
}
```

8

第14回の内容

関数の応用（再帰呼び出し）

第15回 1/14 の小テスト（プログラミング試験）について

9

関数の再帰的な呼び出し

関数（定義）内で自分自身を呼び出しているような関数

```
#include <stdio.h>

int rsum(int n);           rsum() 関数は 1~n までの自然数の和を返す関数

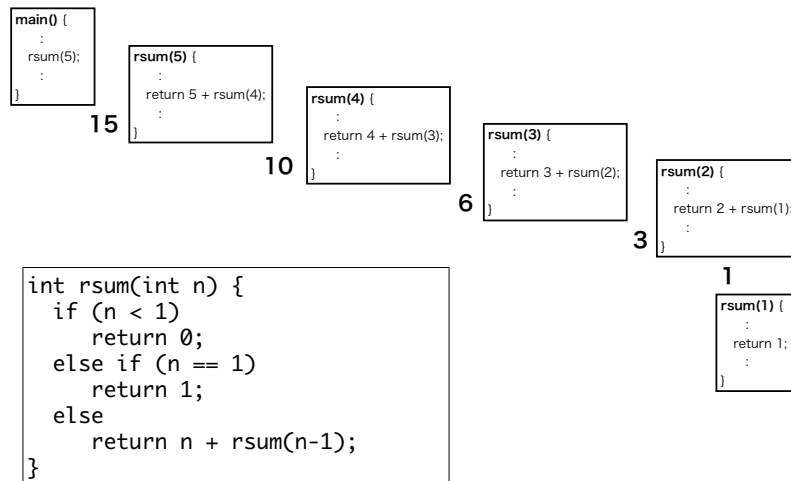
int main(void) {
    int n;
    scanf("%d", &n);
    printf("%d\n", rsum(n));    1から入力された n までの和を出力
    return 0;
}

int rsum(int n) {
    if (n < 1)
        return 0;
    else if (n == 1)
        return 1;
    else
        return n + rsum(n-1);    rsum() 内で rsum() 自身を（再帰的に）呼び出し！
}
```

$rsum(5) = 5 + 4 + 3 + 2 + 1 = 5 + rsum(4)$
 $rsum(4) = 4 + rsum(3)$
 $rsum(3) = 3 + rsum(2)$
 $rsum(2) = 2 + rsum(1)$
 $rsum(1) = 1$

10

関数の呼び出しごとに実体が複製される



11

再帰関数の例：ユークリッドの互除法

（少なくとも1つが0でない）2つの自然数 m, n ($m \geq n$)の最大公約数（GCD）を求めるアルゴリズム

1071 と 1029 の GCD は 21

$1071 \% 1029 = 42$

$1029 \% 42 = 21$

$42 \% 21 = 0$

`gcd(1071, 1029);`

`gcd(1029, 1071%1029);`

`gcd(42, 1029%42);`

`gcd(21, 0);`

```
int gcd(int m, int n) {
    if (n == 0)
        return m;
    else
        return gcd(n, m%n);
}
```

12

第14回の内容

関数の応用（再帰呼び出し）

第15回 1/14 の小テスト（プログラミング試験）について

13

プログラミング試験に関するFAQ

選択問題はいくつ解答したらよいですか？

すべて答えても、一つも答えなくても構いません。各問の配点は Sharif-Judge 上で確認できます。提出されたものは全て採点され、その合計点が50点満点で成績に反映されます。

デスクトップ画面のやカメラ画像などの録画に失敗した場合はどうなりますか？

受験したと認められません。受験途中で操作ミスなどにより録画を停止してしまった場合は、速やかに録画を再開し、録画リンクは試験時間中の全ての（複数の）URLリンクを提出すればOKです。特別な事情があると認められた場合は、考慮されることがあります。

自宅のネットワーク環境やPCが不安定で試験がうまく受けられるか心配です。

入構制限が無ければ、大学内の他の実習室などで受験することも可能です。自分のPCの動作に不安がある場合は、大学の実習室PCで動作する Ubuntu Linux を利用することができます。ただし、実習室PCでのデスクトップ録画のために、各自で事前の準備が必要です。また、コロナ感染状況によっては大学の実習室を使えなくなることがあります。学部のサポートルーム（1-443）などを利用して自宅PC環境の整備を進めてください。

試験中にTAからのサポートを受けることができますか？

PC操作や Sharif-Judge への提出を含め、試験中のTAサポートは受けられません。

15

小テスト（プログラミング試験）について

日時：1月14日（木）4-5講時（第15回）最大2時間

場所：オンライン（ネットワーク環境必須）

実施方法：

複数問題の選択制で、50点を超える点数は切り捨て（50点満点）

問題の提示と提出は Sharif-Judge（自動採点はされません）

指定方法（Awesome Screenshot）でのデスクトップ録画が必須

自分のソースファイル、ネットワーク検索、書籍資料の参照可

他者とのコミュニケーション（会話・チャット・SNS・ファイル送付など）不可

PCのデスクトップ画面・PCカメラ画像・音声録画は原則必須

14

プログラミング試験に関するFAQ 2

カメラ動画や音声録音がどうしてもうまくいきません。

特別な事情があると認められる場合には、カメラ動画や音声録音が含まれない場合も受験として認められますので、事前に申し出て下さい。ただし、これらは不正行為が疑われたときの潔白性の証明となりますので、利用を強くお勧めします。

同一のソースファイル（コピーコード）であるという判定はどうやってなされますか？

ソースコードの記述の同一性をチェックした上で、デスクトップ録画でPC上でのプログラミングの過程を確認し、他人のコピーコードであるか、あるいは、自分で作成したものが偶然に他のコードと類似したものを判定します。

コピーコードが疑われた場合はどうなりますか？

上記でコピーコード認定されたものは不正行為として扱われます。また、疑わしいケースの場合は、後日、口頭試問を行い各問の可否を判定します。悪質なものは学則に則って処理します。

その他、試験に関する質問があれば「★質問チャンネル」へ

16