"Ss. Cyril and Methodius" University in Skopje

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Structured programming

Exercises 5

Version 1.0, 10 October, 2016

Table of Contents

1. Loop structures `for`, `while`, `do-while`

1.1. Problem 1a

Write a program that computes the sum of all even two-digit numbers. Print the result on standard output.

Solution

```c
#include <stdio.h>
int main () {
    int i = 10, sum = 0;
    while (i <= 98) {
        sum = sum + i;
        i+=2;
    }
    printf("%d\n", sum);
    return 0;
}
```

1.2. Problem 1b

Write a program that computes the sum of all odd two-digit numbers. Print the result on standard output in the following format:

```
11 + 13 + 15 + 17 + ... + 97 + 99 = 2475
```

> **ⓘ** The program should be written without using `if`

Solution 1

```c
#include <stdio.h>
int main () {
    int i = 11, sum = 0;
    printf("%d", i);
    sum = i;
    i=i+2;
    while (i <= 99){
        printf(" + %d", i);
        sum = sum + i;
        i+=2;
    }
    printf(" = %d\n", sum);
    return 0;
}
```

Solution 2

```c
#include <stdio.h>
int main () {
    int i = 11, sum = 0;
    while (i <= 97) {
        printf("%d + ", i);
        sum = sum + i;
        i+=2;
    }
    printf(" %d", i);
    sum = sum + i;
    printf(" = %d\n", sum);
    return 0;
}
```

1.3. Problem 2

Write a program that computes на y = x^n for given natural number n, n >= 1 and real number x.

Solution 1

```c
#include <stdio.h>
int main () {
    int counter = 0, n;
    float x, y = 1;
    printf("x: ");
    scanf("%f", &x);
    printf("n: ");
    scanf("%d", &n);
    while (counter < n) {
        y *= x;
        counter++;
    }
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}
```

Solution 2

```c
#include <stdio.h>
int main () {
    int counter = 0, n;
    float x, y = 1;
    printf("x: ");
    scanf("%f", &x);
    printf("n: ");
    scanf("%d", &n);
    do {
        y *= x;
        counter++;
    } while (counter < n);
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}
```

Solution 3

```c
#include <stdio.h>
int main () {
    int counter, n;
    float x, y;
    printf("x: ");
    scanf("%f", &x);
    printf("n: ");
    scanf("%d", &n);
    for(counter = 1, y = x; counter < n; counter++) {
        y *= x;
    }
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}
```

Solution 4

```c
#include <stdio.h>
int main () {
    int counter = 0, n;
    float x, y = 1;
    printf("x: "); scanf("%f", &x);
    printf("n: "); scanf("%d", &n);
    for( ; counter < n; counter++) {
        y *= x;
    }
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}
```

1.4. Problem 3

Write a program that for n numbers read from SI will count the numbers divisible by 3, have residue 1, and have residue 2.

ⓘ   Solve the problem using `while`, `do…while` and `for`

Solution with `while`:

```c
#include <stdio.h>
int main () {
    int n = 1, i = 0, number, div, r1, r2;
    div = r1 = r2 = 0; /*  counters */
    scanf("%d", &n);    /* numbers read from SI */
    while (i < n) {
        scanf("%d", &number);
        if (number % 3 == 0)
            div++;
        else if (number % 3 == 1)
            r1++;
        else r2++;
        i++;
    }
    printf("%d\n", div);
    printf("%d\n", r1);
    printf("%d\n", r2);
    return 0;
}
```

Solution with `do… while`*:*

```c
#include <stdio.h>
int main () {
    int n = 1, i = 0, number, div, r1, r2;
    div = r1 = r2 = 0; /*   counters */
    scanf("%d", &n);    /* numbers read from SI */
    do {
        scanf("%d", &number);
        if (number % 3 == 0)
            div++;
        else if (number % 3 == 1)
            r1++;
        else r2++;
        i++;
    } while (i < n);
    printf("%d\n", div);
    printf("%d\n", r1);
    printf("%d\n", r2);
    return 0;
}
```

Solution with `for`

```c
#include <stdio.h>
int main () {
    int n = 1, i = 0, number, div, r1, r2;
    div = r1 = r2 = 0; /*   counters */
    scanf("%d", &n);    /* numbers read from SI */
    for (i = 0; i < n; ++i) {
        scanf("%d", &number);
        if (number % 3 == 0)
            div++;
        else if (number % 3 == 1)
            r1++;
        else r2++;
    }
    printf("%d\n", div);
    printf("%d\n", r1);
    printf("%d\n", r2);
    return 0;
}
```

1.5. Problem 4

Write a program that will print all 4-digit numbers in which the sum of the three least significant digits is equal to the most significant digit.

4031 (4=0+3+1), 5131 (5=1+3+1)

Solution

```c
#include <stdio.h>
int main() {
    int i, n, sum, first_digit, digit;
    i = 1000;
    while (i <= 9999) {
        first_digit = i / 1000;
        n = i % 1000;
        sum = 0;
        while (n > 0) {
            digit = n % 10;
            sum += digit;
            n /= 10;
        }
        if (sum == first_digit) printf("%d\t", i);
        i++;
    }
    return 0;
}
```

1.6. Problem 5

Write a program that will print all numbers in given range which are read the same from left to right and opposite.

example numbers

```
12321 5061605
```

Solution

```c
#include <stdio.h>
int main () {
    int i, from, to, temp, op, digit;
    scanf("%d %d", &from, &to);
    for (i = from; i <= to; i++) {
        temp = i;
        op = 0;
        while (temp > 0) {
            digit = temp % 10;
            op = op * 10 + digit;
            temp /= 10;
        }
        if (op == i) printf("%d\t", i);
    }
    return 0;
}
```

1.7. Problem 6

Write a program that for unknown count of integers read from SI will find the number with maximum value. The program stops when the reading of integer fails.

Solution

```c
#include <stdio.h>
int main() {
    int n, max;
    if (scanf("%d", &max)) {
        while (scanf("%d", &n)) {
            if (max < n) {
                max = n;
            }
        }
        printf("%d", max);
    } else {
        printf("This is not a number.");
    }
    return 0;
}
```

1.8. Problem 7

Write a program that for unknown count of integers read from SI will find the number with maximum value. Numbers larger than 100 should be ignored. The program stops when the reading of integer fails.

Solution

```c
#include <stdio.h>
int main() {
    int n, max;
    if (scanf("%d", &max)) {
        while (scanf("%d", &n)) {
            if (n > 100) continue;
            if (max < n) {
                max = n;
            }
        }
        printf("%d", max);
    } else {
        printf("This is not a number.");}
    return 0;
}
```

1.9. Problem 8

Write a program that for unknown count of integers read from SI will find the two with maximum values.

example

For the numbers 2 4 7 4 2 1 8 6 9 7 10 3 the program should print 10 and 9.

Solution

```c
#include <stdio.h>
int main() {
    int n, max1, max2, temp;
    if (scanf("%d%d", &max1, &max2) == 2) {
        if (max2>max1){
            temp = max1;
            max1 = max2;
            max2 = temp;
        }
        while(scanf("%d", &n)) {
            if(n > max1){
                max2 = max1;
                max1 = n;
            } else if (n>max2) {
                max2 = n;
            }
        }
        printf("%d\n", max1);
        printf("%d\n", max2);
    } else {
        printf("Enter at least 2 numbers");
    }
    return 0;
}
```

1.10. Problem 9

Write a program that for N integers read from SI will the difference of the sums of numbers on odd and even positions (by the order of reading). If this difference is less than 10 print the message "`The two sums are close`" else print "`The two sums are far`".

example

For numbers:

2 4 3 4 2 1 1 6 1 7

sum_odd_positions = 9

sum_even_positions = 22

Will print:

```
The two sums are far
```

Solution

```c
#include <stdio.h>
int main() {
    int difference, i, n = 0, number = 0;
    int sum_odd_positions = 0, sum_even_positions = 0;
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        scanf("%d", &number);
        if (i % 2) {
            sum_odd_positions += number;
        } else {
            sum_even_positions += number;
        }
    }
    difference = sum_even_positions - sum_odd_positions;
    if (difference < 10 && difference > -10) { //if(abs(difference) < 10){
        printf("The two sums are close");
    } else {
        printf("The two sums are far");
    }
    return 0;
}
```

1.11. Problem 10

Write a program that for unknown count of integers read from SI will find the positions of the successive numbers with maximum sum. The program stops when two successive read numbers are negative.

Solution

```c
#include <stdio.h>
int main() {
    int pol_position, position, max_sum, sum, previous, next;
    scanf("%d%d", &previous, &next);
    pol_position = position = 2;
    max_sum = sum = previous + next;
    while (1) {
        if (previous < 0 && next < 0) {
            break;
        }
        sum = previous + next;
        if (sum > max_sum) {
            max_sum = sum;
            pol_position = position;
        }
        previous = next;
        scanf("%d", &next);
        position++;
    }
    if (position > 2)
        printf("Numbers are found on positions %d and %d with sum: %d\n",
                pol_position - 1, pol_position, max_sum);
    return 0;
}
```

2. The `switch` expression

2.1. Example

```c
#include <stdio.h>
int main() {
    char answer;
    printf("Is SP an easy course? (Y/N): ");
    //answer=getchar();
    scanf("%c", &answer);

    switch (answer) {
        case 'Y':
        case 'y': printf("I think so too!\n");
            break;
        case 'N':
        case 'n': printf("Really?\n");
            break;
        default:
            printf("Is this Yes or No?\n");
    }

    return 0;
}
```

2.2. Problem 1

Write a program that for a read character from SI will print if it is a vowel, digit or something else.

Solution

```c
#include <stdio.h>
int main() {
    char character;
    printf("Enter character: ");
    character = getchar(); /* scanf("%c", &character); */

    switch (character) {
        case 'a': case 'A':
        case 'e': case 'E':
        case 'i': case 'I':
        case 'o': case 'O':
        case 'u': case 'U':
            printf("Vowel is entered: %c\n", character);
            break;
        case '0': case '1': case '2': case '3': case '4':
        case '5': case '6': case '7': case '8': case '9':
            printf("Digit is entered: %c\n", character);
            break;
        default:
            printf("Character is entered: %c\n", character);
    }

    return 0;
}
```

2.3. Problem 2

Write a program for simple calculator. The program should read two numbers and operator in the following format:

```
num1 operator num2
```

After the operation depending on the sign, the result is printed in the following format:

```
num1 operator num2 = result
```

Solution

```c
#include <stdio.h>
int main() {
    char op;
    float num1, num2, result = 0;
    printf("num1 operator num2\n");
    scanf("%f %c %f", &num1, &op, &num2);
    switch (op) {
    case '+':
        result = num1 + num2;
        break;
    case '-':
        result = num1 - num2;
        break;
    case '*':
        result = num1 * num2;
        break;
    case '/':
        if (num2 == 0) {
            printf("Division with 0\n");
        } else {
            result = num1 / num2;
        }
        break;
    default:
        printf("Unknown operator %c\n", op);
        break;
    }
    if (result) printf("%.2f %c %.2f = %f", num1, op, num2, result);
    return 0;
}
```

2.4. Problem 3

Write a program that will compute the average of a student, and the number of failed exams. The number of taken exams by the student is not known.

Solution 1

```c
#include <stdio.h>
int main() {
    int grade, sum = 0, num_grades = 0, failed = 0;
    while (scanf("%d", &grade) == 1) {
        while (grade < 5 || grade > 10) {
            printf("Enter grade from 5 to 10: ");
            if (scanf("%d", &grade) != 1) break;
        }
        /* counting */
        if (grade >= 6 && grade <= 10) {
            sum += grade;
            num_grades++;
        }
        else /*if (grade == 5) */
            failed++;
    }
    if (num_grades == 0)
        printf("No passed exams");
    else
        printf("Average is %4.2f\n", (float)sum / num_grades);
    if (failed != 0)
        printf("Failed on %d exams\n", failed);
    else
        printf("No failed exams\n");
    return 0;
}
```

Solution 2

```c
#include <stdio.h>
int main() {
    int grade, sum = 0, num_grades = 0, failed = 0;
    while (scanf("%d", &grade) == 1) {
        switch (grade) {
            case 6:
            case 7:
            case 8:
            case 9:
            case 10: sum += grade;
                     num_grades++; break;
            case 5:  failed++; break;
            default: printf("Enter grade from 5 to 10: "); break;
        }
    }

    if (num_grades == 0)
        printf("No passed exams");
    else
        printf("Average is %4.2f\n", (float)sum / num_grades);
    if (failed != 0)
        printf("Failed on %d exams\n", failed);
    else
        printf("No failed exams\n");
    return 0;
}
```

2.5. Example of text menu (Command Line Interface - CLI)

Solution

```c
#include <stdio.h>
int main() {
    int value, choice;
    printf("Enter initial value: ");
    scanf("%d", &value);

    do {
        do {
            printf("Menu:\n");
            printf("1 - increase\n");
            printf("2 - decrease\n");
            printf("3 - double\n");
            printf("0 - END\n");
            printf("choice : ");
            scanf("%d", &choice);
        } while ((choice < 0) || (choice > 3));
        switch (choice) {
            case 1: value++; break;
            case 2: value--; break;
            case 3: value *= 2; break;
            case 0: printf("End\n");           break;
            default: printf("Invalid choice!\n"); break;
            }
            printf("value = %d\n", value);
        }
        while (choice != 0);
    return 0;
}
```

3. Practice problems

3.1. Practice problem 1

Given x = sin y with initial value `0.02` up to `0.80` with step `0.01`. Write a program that will print table of y in degrees, minutes and seconds.

y=arctan(\frac{x}{\sqrt{1 - x^2}})

Solution

```c
#include <stdio.h>
/* #define _USE_MATH_DEFINES */
#include <math.h>
int main() {
    float x, y;
    int step, min, sec;
    printf("  x\tdegrees\tminutes\tseconds\n");
    for (x = 0.02; x <= 0.8; x += 0.01) {
        y = 180/M_PI * atan(x / sqrt(1 - x*x));
        step = y;
        min = (y - step) * 60;
        sec = ((y - step) * 60 - min) * 60 + 0.5;
        printf("%5.2f\t%3d\t %2d\t %2d\n", x, step, min, sec);
    }
    return 0;
}
```

3.2. Practice problem 2

Write a program for computing approximation of \pi using the series of

Gregory–Leibniz

1 − \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} − \ldots= \frac{\pi}{4}

\sum_{n=0}^{\infty}\frac{(-1)^n}{2n + 1} = \frac{\pi}{4}

- using the first 100 members.

- until the absolute value of the member that is adding is less then 10e-6

Solution 1.1

```c
#include <stdio.h>
#define _USE_MATH_DEFINES
#include <math.h>
int main() {
    double member = 1, pi = 0;
    int i, sign = 1, denominator = 1;
    for(i=0; i<100; i++) {
        pi += member;
        denominator += 2;
        sign = -sign;
        member = (double)sign / denominator;
    }
    pi*=4;
    printf("pi (approximate) = %lf\n", pi);
    return 0;
}
```

Solution 1.2

```c
#include <stdio.h>
#define _USE_MATH_DEFINES
#include <math.h>
int main() {
    double pi = 0, znak = 4, clen = znak;
    int imenitel = 1;
    do {
        pi += clen;
        imenitel += 2;
        znak = -znak;
        clen = znak / imenitel;
    } while (fabs(clen) > 1e-6);

    /*  pi = clen;
        while (fabs(clen) > 1e-6)
            pi += clen = ((znak*=-1) / (imenitel+=2));*/

    printf("pi (priblizno) = %10.8lf \t pi = %10.8lf\n", pi, M_PI);

    return 0;
}
```

Solution 2

```c
#include <stdio.h>
#define _USE_MATH_DEFINES
#include <math.h>
int main() {
    int faktori = 1, faktorb = 1;
    double pi = 1, clen = 1;
    /* vo ciklusot se presmetuva pi/2 */
    do {
        faktori += 2;
        clen *= (double)faktorb / faktori;
        faktorb++;
        pi += clen;
    } while (clen > 1e-6);


    /*  do
            pi += clen*= (double)faktorb++ / (faktori+=2);
        while (clen > 1e-6); */

    pi *= 2;
    printf("pi (priblizno) = %10.8lf \t pi = %10.8lf\n", pi, M_PI);

    return 0;
}
```

3.3. Practice problem 3

Write a program for approximate computation of \pi using the expression
4*\pi=arctan(1) and continuation of arctan into infinite series.

Solution

```c
#include <stdio.h>
#include <math.h>
int main()
{
    int faktori = 1, faktorb = 1;
    double pi = 1, clen = 1;
    /* vo ciklusot se presmetuva pi/2 */
    do
    {
        faktori += 2;
        clen *= (double)faktorb / faktori;
        faktorb++;
        pi += clen;
    } while (clen > 1e-6);


    /*    do
            pi += clen*= (double)faktorb++ / (faktori+=2);
        while (clen > 1e-6); */

    pi *= 2;
    printf("pi (priblizno) = %10.8lf \t pi = %10.8lf\n", pi, M_PI);

    return 0;
}
```

4. Source code of the examples and problems

https://github.com/finki-mk/SP/

Source code ZIP