

Wykonał: Radosław Smoter

Grupa: 14

Nr: 27

Numer zadania: 4

Przykład: 62

Prowadzący: Prof. dr hab. inż.
Volodymyr Samotyy

Politechnika Krakowska

Wydział Inżynierii Elektrycznej i Komputerowej

Sprawozdanie: Wstęp do Programowania

Spis treści

Polecenie.....	1
Kod programu.....	2
Wyniki.....	5
Opis Programu.....	6
Wnioski.....	7

Polecenie

Obliczyć wartość funkcji jednoargumentowej. Z wartości y wybieramy dwie (trzy) cyfry i wyświetlamy wynik w systemach: dwójkowym, ósemkowym i szesnastkowym. Dla poszczególnych zadań wybieramy cyfry zgodnie z tabelą.

Zadanie	Cyfry
62	3, 5, 6

Funkcja: $y = \frac{5}{3 + \lg^2(x)}$.

Dziedzina: $x = 1.2$.

Kod programu

```

/**
 * @file main.c
 * @author Radosław Smoter (radoslaw.smoter@student.pk.edu.pl)
 * @version 0.1
 * @date 2021-12-16
 *
 * @copyright Copyright (c) 2021
 */

/*
 * Calculate value of the given function.
 * Select 3 digits {3, 5, 6} from the function output and display them in different
counting systems: binary, octal, hexadecimal.
*/

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

double doMath(double);
void showValues(double);

int main(void)
{
    /* Domain */
    double x = 1.2;

    /* Result of the mathematical function */
    double value = doMath(x);

    /* If the mathematical function resulted in non-number value, it can't be
proceeded, because it doesn't have any digits to show. */
    if (value == NAN) return -1;

    showValues(value);

    return 0;
}

/* Calculate a mathematical function */
double doMath(double x) {
    double log1 = log(x);

    /* NAN exception */
    if (isnan(log1)) return NAN;

    /* Real value */
    return 5 / (3 + pow(log1 / log(10), 2));
}

unsigned short int selectDigit(double n, int k) {
    /* Format number into scientific notation without maintaining the powerset e.g.
abc.def --> a.bcdef * 10^0 <=> a.bcdef */
    while (n < 0)

```

```

    n *= 10;
while (n > 10)
    n /= 10;

/* Transform from form a.bcdef into a.bcdef * 10^(k-1) e.g. for the first digit it
is a.bcdef * 10^0; for the fifth, a.bcdef * 10^4 */
for (int i = 0; i < k - 1; i++)
    n *= 10;

/* Cast n into integer; return wanted digit e.g. (a.bcdef * 10^0) % 10 = a;
(a.bcdef * 10^4) % 10 = e */
return (int)n % 10;
}

/* If n is even - return 0; otherwise - return 1 */
_Bool isEven(int n) {
    return n % 2;
}

/* Convert an integer to binary form and print it to stdout */
void decToBin(unsigned long int n)
{
    /* Remainders of division (n % 2) */
    int remainder[65];
    /* Current index */
    int i = 0;

    do {
        remainder[i] = isEven(n);
        i++;
    } while ((n /= 2) > 0);

    /* Determine how many zeros to add to remainder to maintain format in eights with
zeros in the beginning */
    while(i % 8)
    {
        remainder[i] = 0;
        i++;
    }

    /* Binary form of n */
    char bin[65];
    /* Flip remainder. Add '0' to every integer so it becomes it's corresponding
character */
    for (int j = 0; j < i; j++)
        bin[j] = remainder[i - j - 1] + '0';

    /* End as string */
    bin[i] = '\0';
    /* Print formatted version */
    printf("%10s", bin);
}

/* Show selected digits of a given value in bin, oct, hex */
void showValues(double val) {

    /* Number of digit to choose from val (from left) */
    unsigned short digitPlace[] = { 3, 5, 6 };
    /* Chosen digits */
    unsigned short digits[3];

```

```

/* Choose digits from val */
for (int i = 0; i < 3; i++) {
    digits[i] = selectDigit(val, digitPlace[i]);
}

/* Full value */
printf("%10s%20.12f\n", "Value:", val);

/* Counting system */
printf("%10s%10s%10s\n", "bin", "oct", "hex");

for (int i = 0; i < sizeof(digitPlace) / sizeof(short); i++)
{
    /* As bin */
    decToBin(digits[i]);
    /* As oct */
    printf("%10o", digits[i]);
    /* As hex */
    printf("%10X", digits[i]);

    printf("\n");
}
}

```

Wyniki

Value:	1.663190781037	
bin	oct	hex
00000110	6	6
00000001	1	1
00001001	11	9

Opis Programu

W ciele funkcji wykonuje się funkcja zwracająca wynik żądanej operacji matematycznej, następnie sprawdzany jest wynik tej funkcji – jeśli zwróci NAN, to program zakończy swoje działanie. W przeciwnym przypadku wykona się funkcja showValues(), która pokaże cyfry o określonych numerach (w liczbie), w systemach: binarnym, ósemkowym i heksadecymalnym.

Funkcja doMath() oblicza wartość funkcji matematycznej określonej w zadaniu. Przy operacjach niebezpiecznych (takaich, których dziedziny nie są całym przedziałem liczb rzeczywistych), sprawdza, czy wynik zwraca NAN. Jeśli tak – kończy działanie funkcji i zwraca NAN; w przeciwnym przypadku zwraca wartość operacji.

Funkcja selectDigit() wybiera k-tą cyfrę liczby n. Najpierw sanitaryzuje input, czyli konwertuje go do postaci notacji naukowej. Następnie przesuwając miejsce dziesiętne liczby n do (k - 1) pozycji w prawo. Zwraca resztę z dzielenia poprzedniej operacji przez 10, jednocześnie konwertując poprzedni wynik do liczby całkowitej.

Funkcja decToBin() zamienia podaną w parametrze wartość całkowitą n na liczbę w postaci binarnej. Zapisuje resztę z dzielenia n przez 2 do tablicy remainder[] przy każdej iteracji dzieląc liczbę przez 2, aż liczba wyniesie zero. Następnie dodaje zera nieznaczące do końca tablicy remainder[], aż liczba utrzyma postać w wielokrotnościach liczby osiem. Dalej tablica remainder[] zostaje odwrócona do tablicy bin[] i zakończona przez '\0', gdzie staje się napisem. Funkcję kończy wypisanie wartości napisu bin do standardowego wyjścia (stdout).

Funkcja showValues() pokazuje 3, 5 i 6 cyfrę podanej wartości val w systemach binarnym, ósemkowym i heksadecymalnym na standardowe wyjście (stdout). Do tablicy digits[] zapisuje cyfry o wybranych numerach. Następnie wypisuje w sformatowany sposób wartość liczby val dla referencji, a później wartości wybranych cyfr w określonych systemach liczenia, które są opisane skrótowo, dla kolumn. Wyniki systemów ósemkowego i szesnastkowego otrzymywane są dzięki specyfikatorom funkcji printf(), tj. "x" – sys. szesnastkowy, "o" – sys. ósemkowy. Wynik dla systemu binarnego podaje funkcja decToBin().

Wnioski

Wyrażenie matematyczne jest poprawnie obliczone.

Value: 1.663190781037



5 / (3 + lg(x)^2) for 1.2



NATURAL LANGUAGE



MATH INPUT



EXTENDED KEYBOARD



EXAMPLES



UPLOAD



RANDOM

Assuming "lg" is the base 10 logarithm | Use [the natural logarithm](#) or [the base 2 logarithm](#) instead

Input interpretation

$$\frac{5}{3 + \log_{10}^2(x)} \text{ where } x = 1.2$$

Result

1.66319

Na tej podstawie można stwierdzić, że przedstawione liczby również są poprawnie wybrane.