

SPRAWOZDANIE		Data wykonania: 16/01/2019
Tytuł zadania:	Wykonał:	Sprawdził:
<i>Maszyna Turinga</i>	<i>Krzysztof Maciejewski</i> 299262	<i>dr inż. Konrad Markowski</i>

Spis treści

1.	Cel ćwiczenia	2
2.	Teoria.....	2
3.	Szczegóły implementacyjne	3
4.	Sposób wywołania programu.....	5
4.1.	Wybór opcji nr. 1	6
4.2.	Wybór opcji nr. 2	6
4.3.	Wybór opcji nr.3	7
4.4.	Wybór opcji nr. 4	7
4.5.	Błędne dane w pliku do odczytu	8
4.6.	Nie można otworzyć pliku	8
4.7.	Błędny wybór opcji	8
4.8.	Nie podano dodatniej liczby całkowitej	8
4.9.	Pierwsza podana liczba jest mniejsza od drugiej	8
5.	Wnioski i spostrzeżenia	9

1. Cel ćwiczenia

Celem ćwiczenia było napisanie emulatora maszyny Turinga obliczającej różnicę właściwą. Program miał wyświetlać opis MT oraz ciąg opisów chwilowych dla zadanej taśmy. Dodatkowo, program powinien zawierać menu, z którego można wybrać opcję odczytu liczb z pliku oraz zapisu wyniku do pliku.

Zadanie 23. Napisać emulator maszyny Turinga obliczającą różnicę właściwą:

$$m - n = \begin{cases} m - n & \text{dla } m \geq n \\ 0 & \text{dla } m < n \end{cases}$$

dla parametrów zakodowanych unarnie.

Postać MT

$$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, 0)$$

dla

δ	0	1	B
q_0	(q_1, B, P)	(q_5, B, P)	-
q_1	$(q_1, 0, P)$	$(q_2, 1, P)$	-
q_2	$(q_3, 1, L)$	$(q_2, 1, P)$	(q_4, B, L)
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	(q_0, B, P)
q_4	$(q_4, 0, L)$	(q_4, B, L)	$(q_6, 0, P)$
q_5	(q_5, B, P)	(q_5, B, P)	(q_6, B, P)
q_6	-	-	-

Program powinien:

- Wyświetlić opis MT.
- Dla wczytanych dwóch liczb całkowitych generować taśmę wejściową zakodowaną unarnie.
- Wyświetlać ciąg opisów chwilowych MT dla zadanej taśmy wejściowej.
- Po zatrzymaniu automatu zinterpretować otrzymany wynik.

2. Teoria

Maszyna Turinga to matematyczny model obliczeń, który definiuje abstrakcyjną maszynę służącą do wykonywania algorytmów.

Formalnie maszynę Turinga możemy przedstawić jako **uporządkowaną siódemkę**:

$$MT = \langle Q, \Sigma, \Gamma, \delta, q_0, B, F \rangle,$$

gdzie:

Q to skończony zbiór stanów,

Σ to zbiór symboli wejściowych,

Γ to zbiór dopuszczalnych symboli,

δ to funkcja przejścia,

q_0 to stan początkowy,

B to symbol pusty,

F to zbiór stanów końcowych.

Działanie:

MT składa się z bloku sterowania, głowicy odczytującej i zapisującej oraz nieskończonej długiej taśmy. Każda komórka taśmy zawiera jeden symbol. Głowica zawsze jest ustawiona nad jedną z komórek, zaś maszyna znajduje się w jednym z określonych stanów. Zależnie od kombinacji stanu maszyny i symbolu napotkanego na taśmie, maszyna zmienia stan, a głowica zapisuje nową wartość w polu oraz przesuwa się jedną komórkę w prawo/lewo lub nie zmienia położenia.

Działanie MT określa funkcja przejścia, którą można przedstawić w formie tablicy przejść.

3. Szczegóły implementacyjne

Tablica przejść MT została zapisana w programie w postaci trzech tablic dwuwymiarowych. Tablica *stany* określa zmianę stanu, *symbol* zmianę znaku na taśmie, zaś *przejscie* przesunięcie głowicy. Pierwszy wymiar tablicy to zależność od aktualnego stanu maszyny (q_0 – q_6), zaś drugi wymiar to zależność od wczytanego znaku (0, 1 lub B).

```
//funkcje przejść zapisujemy jako tablicę dwuwymiarową
int stany[7][3] = { {1, 5, -1}, {1, 2, -1}, {3, 2, 4}, {3, 3, 0}, {4, 4, 6}, {5, 5, 6}, {-1, -1, -1} };
char symbol[7][3] = { {'B', 'B', '-'}, {'0', '1', '-'}, {'1', '1', 'B'}, {'0', '1', 'B'}, {'0', 'B', '0'}, {'B', 'B', 'B'}, {'-', '-', '-'} };
char przejscie[7][3] = { {'P', 'P', '-'}, {'P', 'P', '-'}, {'L', 'P', 'L'}, {'L', 'L', 'P'}, {'L', 'L', 'P'}, {'P', 'P', 'P'}, {'-', '-', '-'} };
```

Z racji, że tablica stanów jest typu int, brak rozkazu został przedstawiony jako wartość -1.

Program wyświetla menu, którego możemy wybrać jedną z czterech opcji.

```
//opis MT
printf("*****\n");
printf("*\n");
printf("*          Emulator MT obliczający różnicę właściwą          *\n");
printf("*\n");
printf("*      M = ({q0, q1, q2, q3, q4, q5, q6}, {0, 1}, {0, 1, B}, δ, q0, B, 0)      *\n");
printf("*\n");
printf("*      Dostępne opcje:\n");
printf("*      1) Odczytaj z pliku, zapisz do pliku\n");
printf("*      2) Odczytaj z pliku, wyświetl na ekran\n");
printf("*      3) Odczytaj z ekranu, zapisz do pliku\n");
printf("*      4) Odczytaj z ekranu, wyświetl na ekran\n");
printf("*\n");
printf("*****\n");
```

Program pobiera wybraną przez nas opcję i uruchamia konkretne warunki, w zależności której opcję wybraliśmy.

Jeżeli pierwsza wczytana liczba jest mniejsza od drugiej, to program wyświetli wynik 0, zgodnie z założeniami zadania.

```
if (m < n) {
    printf("\nPierwsza liczba jest mniejsza od drugiej liczby\n");
    printf("Wynik = 0\n");
    return 0;
}
```

Taśmę przedstawimy jako tablicę typu char.

```
char *tasma = tasm_wej (m, n); //generujemy tasme wejsciowa
```

Do wygenerowania taśmy wejściowej korzystamy z funkcji *tasm_wej*.

```

char *tasm_wej(int m, int n){//generowanie taśmy wejściowej

    int dlug = m + n + 2; //dodatkowe 2 miejsca dla 1 i B
    char *tasma = malloc (dlug * sizeof(char*));
    int i=0;

    while (i != m){//przedstawiamy pierwszą liczbę na taśmie w postaci zer
        tasma[i] = '0';
        i++;
    }

    tasma[i] = '1';//rozdzielamy liczby jedyneką
    i++;

    while (i != m + n + 1){//druga liczba w postaci zer
        tasma[i] = '0';
        i++;
    }

    tasma[i] = 'B';//symbol pusty na koniec

    return tasma;
}

```

Funkcja ta przedstawia obie liczby w postaci zer na taśmie, rozdzielonych jedyneką, i umieszcza symbol pusty na końcu.

```
int stan = 0, glowica = 0, err = 0;
```

Zmienna *err* służy do rozpoznania kiedy zakończyć pętlę, zmienna typu int *glowica* wskazuje na znak taśmy, zaś zmienna *stan* przechowuje aktualny stan maszyny jako liczbę od 0 do 6.

Działanie naszej maszyny Turinga symulujemy przy pomocy pętli *while*.

```

while (err != 1){

    switch (tasma[glowica]){//zależnie od przeczytanego znaku

        case '0': if (stany[stan][0] == -1){
                    err = 1;
                    break;
                }
                tasma[glowica] = symbol[stan][0];//zmieniamy znak na tasme
                if(przejscie[stan][0]=='P') glowica += 1;
                else glowica -= 1;//przesuwamy głowice
                tmp = stany[stan][0];
                stan = tmp;//zmieniamy stan
                break;
    }
}

```

W pętli mamy zawartą kolejną pętlę *switch*, która uruchamia konkretny warunek w zależności od wczytanego znaku z taśmy.

Jeżeli program rozpozna, że nie ma następnego rozkazu (następny stan to -1) do zmiennej *err* zostanie przekazana wartość -1 i pętla zostanie przerwana. W innym wypadku nastąpi zmiana znaku, zmiana stanu oraz przesunięcie głowicy, zgodnie z naszą tablicą przejść.

Po pętli *switch* wyświetlamy aktualną taśmę, pozycję głowicy oraz stan maszyny.

```

if (err != 1){ //wypisujemy aktualny stan rzeczy
    fprintf(output, "%s          Pozycja głowicy: %d Stan: %d\n", tasma, glowica, stan);
}

```

Po zakończeniu pętli *while* zliczamy zera i wyświetlamy różnicę.

```
//liczymy zera i wyświetlamy różnicę
fprintf(output, "Różnica wynosi %d\n", policz_zera(tasma));
```

Do zliczania zer na taśmie korzystamy z funkcji *policz_zera*.

```
int policz_zera(char *tasma){//zliczanie zer na taśmie

    int i=0, j=0;

    while (tasma[i] != '\0'){

        if (tasma[i] == '0') j++;
        i++;
    }

    return j;
}
```

4. Sposób wywołania programu

Przy wywoływaniu programu nie podajemy żadnego argumentu. Po wywołaniu pojawi nam się menu, z którego możemy wybrać jedną z czterech opcji odczytu liczb i zapisu wyniku. Wyboru dokonujemy wpisując cyfrę 1, 2, 3 lub 4.

```
*****
*
*          Emulator MT obliczający różnicę właściwą          *
*
*      M = ({q0, q1, q2, q3, q4, q5, q6}, {0, 1}, {0, 1, B}, δ, q0, B, 0)
*
*      Dostępne opcje:
*      1) Odczytaj z pliku, zapisz do pliku
*      2) Odczytaj z pliku, wyświetl na ekran
*      3) Odczytaj z ekranu, zapisz do pliku
*      4) Odczytaj z ekranu, wyświetl na ekran
*
*****
Wybierz opcje: 4
```

Następnie, zależnie od wybranej opcji, program poprosi nas o podanie nazw plików, bądź ręczne podanie liczb.

Przy wyborze opcji odczytu liczb z pliku, ważne jest by plik ten zawierał jedynie 2 liczby całkowite. W innym wypadku program zwróci błąd.

Przykłady wywołania programu:

4.1. Wybór opcji nr. 1

Wybierz opcje: 1

Podaj nazwę pliku do odczytu: 1

Podaj nazwę pliku do zapisu: 2

```
root@kristoph:~/turing# cat 2
```

Wczytano 3 i 1

00010B	Pozycja głowicy: 0	Stan: 0
B0010B	Pozycja głowicy: 1	Stan: 1
B0010B	Pozycja głowicy: 2	Stan: 1
B0010B	Pozycja głowicy: 3	Stan: 1
B0010B	Pozycja głowicy: 4	Stan: 2
B0011B	Pozycja głowicy: 3	Stan: 3
B0011B	Pozycja głowicy: 2	Stan: 3
B0011B	Pozycja głowicy: 1	Stan: 3
B0011B	Pozycja głowicy: 0	Stan: 3
B0011B	Pozycja głowicy: 1	Stan: 0
BB011B	Pozycja głowicy: 2	Stan: 1
BB011B	Pozycja głowicy: 3	Stan: 1
BB011B	Pozycja głowicy: 4	Stan: 2
BB011B	Pozycja głowicy: 5	Stan: 2
BB011B	Pozycja głowicy: 4	Stan: 4
BB01BB	Pozycja głowicy: 3	Stan: 4
BB0BBB	Pozycja głowicy: 2	Stan: 4
BB0BBB	Pozycja głowicy: 1	Stan: 4
B00BBB	Pozycja głowicy: 2	Stan: 6

Różnica wynosi 2

```
root@kristoph:~/turing#
```

4.2. Wybór opcji nr. 2

Wybierz opcje: 2

Podaj nazwę pliku do odczytu: 1

Wczytano 3 i 1

00010B	Pozycja głowicy: 0	Stan: 0
B0010B	Pozycja głowicy: 1	Stan: 1
B0010B	Pozycja głowicy: 2	Stan: 1
B0010B	Pozycja głowicy: 3	Stan: 1
B0010B	Pozycja głowicy: 4	Stan: 2
B0011B	Pozycja głowicy: 3	Stan: 3
B0011B	Pozycja głowicy: 2	Stan: 3
B0011B	Pozycja głowicy: 1	Stan: 3
B0011B	Pozycja głowicy: 0	Stan: 3
B0011B	Pozycja głowicy: 1	Stan: 0
BB011B	Pozycja głowicy: 2	Stan: 1
BB011B	Pozycja głowicy: 3	Stan: 1
BB011B	Pozycja głowicy: 4	Stan: 2
BB011B	Pozycja głowicy: 5	Stan: 2
BB011B	Pozycja głowicy: 4	Stan: 4
BB01BB	Pozycja głowicy: 3	Stan: 4
BB0BBB	Pozycja głowicy: 2	Stan: 4
BB0BBB	Pozycja głowicy: 1	Stan: 4
B00BBB	Pozycja głowicy: 2	Stan: 6

Różnica wynosi 2

```
root@kristoph:~/turing#
```

4.3. Wybór opcji nr.3

Wybierz opcje: 3

Podaj pierwszą liczbę całkowitą: 3

Podaj drugą liczbę całkowitą: 1

Podaj nazwę pliku do zapisu: 2

root@kristoph:~/turing# cat 2

Wczytano 3 i 1

00010B	Pozycja głowicy: 0	Stan: 0
B0010B	Pozycja głowicy: 1	Stan: 1
B0010B	Pozycja głowicy: 2	Stan: 1
B0010B	Pozycja głowicy: 3	Stan: 1
B0010B	Pozycja głowicy: 4	Stan: 2
B0011B	Pozycja głowicy: 3	Stan: 3
B0011B	Pozycja głowicy: 2	Stan: 3
B0011B	Pozycja głowicy: 1	Stan: 3
B0011B	Pozycja głowicy: 0	Stan: 3
B0011B	Pozycja głowicy: 1	Stan: 0
BB011B	Pozycja głowicy: 2	Stan: 1
BB011B	Pozycja głowicy: 3	Stan: 1
BB011B	Pozycja głowicy: 4	Stan: 2
BB011B	Pozycja głowicy: 5	Stan: 2
BB011B	Pozycja głowicy: 4	Stan: 4
BB01BB	Pozycja głowicy: 3	Stan: 4
BB0BBB	Pozycja głowicy: 2	Stan: 4
BB0BBB	Pozycja głowicy: 1	Stan: 4
B00BBB	Pozycja głowicy: 2	Stan: 6

Różnica wynosi 2

root@kristoph:~/turing#

4.4. Wybór opcji nr. 4

Wybierz opcje: 4

Podaj pierwszą liczbę całkowitą: 3

Podaj drugą liczbę całkowitą: 1

Wczytano 3 i 1

00010B	Pozycja głowicy: 0	Stan: 0
B0010B	Pozycja głowicy: 1	Stan: 1
B0010B	Pozycja głowicy: 2	Stan: 1
B0010B	Pozycja głowicy: 3	Stan: 1
B0010B	Pozycja głowicy: 4	Stan: 2
B0011B	Pozycja głowicy: 3	Stan: 3
B0011B	Pozycja głowicy: 2	Stan: 3
B0011B	Pozycja głowicy: 1	Stan: 3
B0011B	Pozycja głowicy: 0	Stan: 3
B0011B	Pozycja głowicy: 1	Stan: 0
BB011B	Pozycja głowicy: 2	Stan: 1
BB011B	Pozycja głowicy: 3	Stan: 1
BB011B	Pozycja głowicy: 4	Stan: 2
BB011B	Pozycja głowicy: 5	Stan: 2
BB011B	Pozycja głowicy: 4	Stan: 4
BB01BB	Pozycja głowicy: 3	Stan: 4
BB0BBB	Pozycja głowicy: 2	Stan: 4
BB0BBB	Pozycja głowicy: 1	Stan: 4
B00BBB	Pozycja głowicy: 2	Stan: 6

Różnica wynosi 2

root@kristoph:~/turing#

4.5. Błędne dane w pliku do odczytu

Wybierz opcje: 1

Podaj nazwę pliku do odczytu: 1

Błędne dane w pliku

Plik powinien zawierać jedynie dwie liczby całkowite

```
root@kristoph:~/turing# cat 1
```

```
3 1b
```

```
root@kristoph:~/turing#
```

4.6. Nie można otworzyć pliku

Wybierz opcje: 1

Podaj nazwę pliku do odczytu: 3

Nie można otworzyć tego pliku

```
root@kristoph:~/turing#
```

4.7. Błędny wybór opcji

Wybierz opcje: 5

Podano niewlasciwa wartosc!

```
root@kristoph:~/turing#
```

4.8. Nie podano dodatniej liczby całkowitej

Wybierz opcje: 4

Podaj pierwszą liczbę całkowitą: b

Nie podano dodatniej liczby całkowitej

```
root@kristoph:~/turing#
```

Wybierz opcje: 4

Podaj pierwszą liczbę całkowitą: -1

Nie podano dodatniej liczby całkowitej

```
root@kristoph:~/turing#
```

4.9. Pierwsza podana liczba jest mniejsza od drugiej

Wybierz opcje: 4

Podaj pierwszą liczbę całkowitą: 1

Podaj drugą liczbę całkowitą: 2

Pierwsza liczba jest mniejsza od drugiej liczby

Wynik = 0

```
root@kristoph:~/turing#
```


5. Wnioski i spostrzeżenia

Napisanie emulatora maszyny Turinga nie było zadaniem prostym. Wymagało ono dobrej znajomości zasad działania MT oraz kreatywnego podejścia przy wielu aspektach programu. Trudność zadania wiązała się również z odpowiednim zabezpieczeniem tak złożonego programu przed wszystkimi możliwymi błędami.

Jestem zadowolony z ogółu mojego programu – jest stosunkowo krótki, lecz również dokładnie zabezpieczony, zaś sam kod jest przejrzysty. Z czego mogę być szczególnie dumny to kontrola błędów odczytu liczb z pliku, którą udało mi się zawrzeć w jednej linii kodu.

Szczególną trudność sprawiło mi napisanie przejść MT w możliwie zwięzły sposób. Jako, że każde przejście zależy od dwóch czynników i wiąże się ze zmianą trzech, rozwiązanie tego problemu wymagało pomysłowości. Ostatecznie udało mi się ograniczyć zawrzenie te przejścia w jednej pętli *switch*, co uważam za sukces.