

Architektura i Organizacja Komputerów II

Lab 9-10

Zadania

Uwaga!

Proszę być świadomym występowania [błędu programu WinDLX](#) przy pisaniu programu. Kod należy napisać w taki sposób, aby nie dopuścić do jego wystąpienia.

Sposób wystawienia oceny

Ocena jest wystawiana na podstawie:

- poprawności wykonanych zadań,
- jakości kodu,
- zrozumienia kodu,
- czytelności kodu,
- samodzielności pracy,
- wykonania wszystkich poleceń w zadaniu.

Pytania do prowadzącego **nie** są traktowane jako praca niesamodzielna.

Dane do zadań

ns: numer stanowiska (na monitorze)

np: numer podany przez prowadzącego na zajęciach

grupa: $g = (ns + np) \bmod 35$, gdzie mod oznacza działanie modulo

Tabela 1.1. Definicja tabeli A

Grupa	Wzór tworzenia elementów dla tabeli A	Rozmiar tabeli A
0	$A[i] = (53-84-(i/3/(g+97)+78*12*(14-i)))$	$n = 93$
1	$A[i] = 69+68+ns+33+5*24*i/ns/(np+g)$	$n = 115$
2	$A[i] = 39-np/(i*89+(27*36*g+83)+76-i)$	$n = 110$
3	$A[i] = np-64-3+i+np*90*(78/51)/15-46$	$n = 118$
4	$A[i] = i/np+66*89-25+ns+25+i+(ns/g)$	$n = 119$
5	$A[i] = (ns-np)/58-g+i-6-g*17*(i/ns)$	$n = 107$
6	$A[i] = g/92+np-ns-(17-12-i/48+(66*g))$	$n = 112$
7	$A[i] = i/87+(i+26+(59*60)/(g*93*(np+i)))$	$n = 84$
8	$A[i] = 39-g-(9-np-i*i*np/25/14+40)$	$n = 112$
9	$A[i] = (g+31+44-21)-i/7/22*ns*(32+71)$	$n = 91$
10	$A[i] = g/np/73+27+(np*ns)*ns-94-(72/i)$	$n = 99$
11	$A[i] = (11-44-ns-i-(g+np+np*ns)*(ns/np))$	$n = 81$
12	$A[i] = (i*ns)*41+g+40-40-i/i+(60/39)$	$n = 95$
13	$A[i] = (33-16)-(64*27*(np/73/np+ns)+i+9)$	$n = 106$
14	$A[i] = np-g-69*g*(70-i-g+g+65/58)$	$n = 88$
15	$A[i] = 25/65+73*ns-19+18*g+35*(i+ns)$	$n = 99$
16	$A[i] = 39-ns-(np-47)-(81-49)-ns/46/(i*np)$	$n = 111$
17	$A[i] = (34-22)-np/45/11*g*(75+ns)+(i+52)$	$n = 100$
18	$A[i] = (ns*54)+i*39-96/ns/i+ns+34-87$	$n = 103$
19	$A[i] = 41*31/52+52+np*53*(i+3)+(67-20)$	$n = 86$
20	$A[i] = 35/i+(ns+52)+89*i*29+i+(i*np)$	$n = 93$
21	$A[i] = 59/i-82*51+33*g*np-i-(38+57)$	$n = 119$
22	$A[i] = i*ns*20-82-(73/np)/(21/ns/(g*43))$	$n = 110$
23	$A[i] = g*np*40+np/72-i-79+ns+(66+np)$	$n = 114$
24	$A[i] = np-44-(89-69)-i*48*(13+ns+(27/i))$	$n = 98$

25	$A[i] = g \cdot np + (i/6) / (np - 75 - 26 - g - g - 80)$	$n = 108$
26	$A[i] = (i - np) - 56 \cdot 72 \cdot 80 / 66 / (i - g) - 43 - g$	$n = 99$
27	$A[i] = (88 \cdot i \cdot 28 + g) \cdot 65 / i / (30 - np - 15 / np)$	$n = 96$
28	$A[i] = (12 \cdot i) / 56 + 89 + 3 / i / 61 \cdot 97 \cdot g - 19$	$n = 97$
29	$A[i] = ns / 93 / g + 96 + 85 + ns + (i \cdot np \cdot 73 + 25)$	$n = 115$
30	$A[i] = 38 - ns - (43 - i - (g \cdot i \cdot g / 12) / (37 + 6))$	$n = 105$
31	$A[i] = ns \cdot g \cdot (75 + 65 / (i + 74 + (ns / 41 / (np + 60))))$	$n = 90$
32	$A[i] = (72 - 68 - (72 + i \cdot (28 + 58) + 82 / g) / (51 \cdot 21))$	$n = 103$
33	$A[i] = i \cdot np / 35 - 12 - (np - 59) - ns \cdot 79 \cdot 49 + np$	$n = 119$
34	$A[i] = i + 76 / 39 + 55 + ns + g + 31 / 24 / (42 \cdot g)$	$n = 93$

Tabela 1.2. Definicja tabeli B.

Grupa	Wzór dla elementów tabeli B	Rozmiar tabeli B
0	$B[i] = A[i+2] + A[i+0] * A[i+5] - A[i+4]$	$m = n - 6$
1	$B[i] = A[i+1] - A[i+6] + A[i+5] * A[i+3]$	$m = n - 6$
2	$B[i] = A[i+1] * A[i+3] + A[i+4] - A[i+2]$	$m = n - 6$
3	$B[i] = A[i+3] * A[i+5] - A[i+6] / A[i+2]$	$m = n - 6$
4	$B[i] = A[i+5] + A[i+0] - A[i+4] / A[i+2]$	$m = n - 6$
5	$B[i] = A[i+4] - A[i+3] / A[i+0] / A[i+5]$	$m = n - 6$
6	$B[i] = A[i+0] / A[i+1] - A[i+4] * A[i+3]$	$m = n - 6$
7	$B[i] = A[i+5] + A[i+2] + A[i+0] + A[i+1]$	$m = n - 6$
8	$B[i] = A[i+0] - A[i+3] * A[i+4] + A[i+2]$	$m = n - 6$
9	$B[i] = A[i+4] + A[i+3] + A[i+6] * A[i+1]$	$m = n - 6$
10	$B[i] = A[i+5] - A[i+1] / A[i+6] - A[i+3]$	$m = n - 6$
11	$B[i] = A[i+3] * A[i+6] - A[i+0] / A[i+4]$	$m = n - 6$
12	$B[i] = A[i+2] / A[i+6] * A[i+1] / A[i+0]$	$m = n - 6$
13	$B[i] = A[i+6] * A[i+5] / A[i+0] + A[i+2]$	$m = n - 6$
14	$B[i] = A[i+5] - A[i+1] / A[i+6] * A[i+2]$	$m = n - 6$
15	$B[i] = A[i+0] + A[i+5] * A[i+6] - A[i+4]$	$m = n - 6$
16	$B[i] = A[i+5] * A[i+4] / A[i+2] * A[i+0]$	$m = n - 6$
17	$B[i] = A[i+2] / A[i+5] + A[i+6] - A[i+1]$	$m = n - 6$
18	$B[i] = A[i+5] - A[i+1] / A[i+2] * A[i+0]$	$m = n - 6$
19	$B[i] = A[i+2] - A[i+3] - A[i+0] * A[i+4]$	$m = n - 6$
20	$B[i] = A[i+1] + A[i+4] - A[i+6] * A[i+0]$	$m = n - 6$
21	$B[i] = A[i+4] - A[i+5] + A[i+2] - A[i+3]$	$m = n - 6$
22	$B[i] = A[i+2] / A[i+0] * A[i+3] + A[i+6]$	$m = n - 6$
23	$B[i] = A[i+0] * A[i+4] + A[i+5] * A[i+1]$	$m = n - 6$
24	$B[i] = A[i+6] / A[i+4] - A[i+1] + A[i+0]$	$m = n - 6$
25	$B[i] = A[i+3] * A[i+6] - A[i+2] * A[i+4]$	$m = n - 6$
26	$B[i] = A[i+6] + A[i+4] / A[i+2] / A[i+0]$	$m = n - 6$
27	$B[i] = A[i+0] + A[i+2] - A[i+4] / A[i+5]$	$m = n - 6$

28	$B[i] = A[i+0] + A[i+3] / A[i+5] - A[i+1]$	$m = n - 6$
29	$B[i] = A[i+6] - A[i+1] - A[i+3] - A[i+5]$	$m = n - 6$
30	$B[i] = A[i+6] + A[i+5] / A[i+0] * A[i+3]$	$m = n - 6$
31	$B[i] = A[i+5] - A[i+2] + A[i+6] + A[i+4]$	$m = n - 6$
32	$B[i] = A[i+6] * A[i+5] + A[i+1] + A[i+4]$	$m = n - 6$
33	$B[i] = A[i+2] / A[i+6] - A[i+4] / A[i+1]$	$m = n - 6$
34	$B[i] = A[i+0] - A[i+4] * A[i+3] - A[i+1]$	$m = n - 6$

Zadanie 7 – tablice, liczby zmiennoprzecinkowe

Napisać program, którego zadaniem będzie utworzenie tablic o wartościach zmiennoprzecinkowych na podstawie podanych wzorów.

1. Utworzyć arkusz kalkulacyjny zawierający obliczone wartości elementów dla tablic A i B zdefiniowanych w [Tabeli 1.1](#) i [Tabeli 1.2](#).
Wszystkie działania mają zostać przeprowadzone jawnie.
Nie dopuszcza się upraszczania wzoru.
Iterator i należy rozpocząć od wartości 0.
2. Napisać program który:
 - 2.1. Utworzy tablicę o etykiecie A zdefiniowaną w [Tabeli 1.1](#).
Wszystkie działania mają zostać przeprowadzone jawnie.
Nie dopuszcza się upraszczania wzoru.
Działania mają zostać wykonane na liczbach zmiennoprzecinkowych pojedynczej precyzji.
Występowanie dzielenia przez zero dla danej iteracji należy zignorować.
 - 2.2. Utworzy tablicę o etykiecie B zdefiniowaną w [Tabeli 1.2](#).
Wszystkie działania mają zostać przeprowadzone jawnie.
Nie dopuszcza się upraszczania wzoru.
Działania mają zostać wykonane na liczbach zmiennoprzecinkowych pojedynczej precyzji.
Występowanie dzielenia przez zero dla danej iteracji należy zignorować.
3. Uruchomić program.
4. Porównać otrzymane wyniki z wartościami z arkusza kalkulacyjnego.
5. Utworzyć nowy dokument tekstowy. Dokument ma zawierać kolejno:
 - 5.1. Wartości zmiennych ns , np , g , n , m .
 - 5.2. Przydzielony wzór z [Tabeli 1.1](#).
 - 5.3. Przydzielony wzór z [Tabeli 1.2](#).
 - 5.4. Zrzut(y) ekranu okna pamięci operacyjnej przedstawiający wartości A .
Wartości mają się znajdować w pojedynczej kolumnie (zmniejszyć szerokość okna pamięci).
 - 5.5. Zrzut(y) ekranu okna pamięci operacyjnej przedstawiający wartości B .
Wartości mają się znajdować w pojedynczej kolumnie (zmniejszyć szerokość okna pamięci).
6. Ustawić okna:
 - 6.1. Z lewej strony ekranu na drugim planie ma znajdować się kod programu.
 - 6.2. Z prawej strony ekranu na drugim planie ma znajdować się program WinDlx.
 - 6.3. Z lewej strony ekranu na pierwszym planie ma znajdować się przygotowany dokument.
 - 6.4. Z prawej strony ekranu na pierwszym planie ma znajdować się przygotowany arkusz kalkulacyjny.
7. Wpisać się na listę: <https://goo.gl/7Y17lh> i czekać na podeście prowadzącego.

Zadanie 8 – scheduling, obliczenia liczby cykli zegarowych

Dokonać optymalizacji programu z *Zadania 7* oraz obliczyć liczbę cykli zegarowych.

1. Nie zamykać arkusza kalkulacyjnego z *Zadania 7*.
2. Utworzyć kopię kodu programu z *Zadania 7*.
3. Uruchomić program.
4. Na podstawie analizy napisanego kodu oraz diagramu cykli zegarowych pisemnie obliczyć liczbę cykli zegarowych, liczbę instrukcji i CPI (Clock cycles Per Instruction)

$$CPI = \frac{\text{liczba cykli zegarowych}}{\text{liczba wykonanych instrukcji}}$$

5. Wykonać i zapisać zrzut ekranu okna statystyk.
6. Wykonać i zapisać zrzuty ekranu okna diagramu cykli zegarowych dla obu pętli.
Upewnić się że na zrzutach ekranu widoczne są nazwy i argumenty instrukcji.
7. **Dokonać optymalizacji kodu tak, aby otrzymać jak najmniejszą liczbę hazardów RAW.**

Należy skorzystać z metody schedulingu.

Nie korzystamy z metody loop unrolling.

Nie dopuszcza się wstawiania instrukcji nop.

8. Uruchomić program.
9. Sprawdzić czy otrzymane wyniki zgadzają się z wartościami z arkusza kalkulacyjnego.
10. Zrobić zrzut ekranu okna statystyk.
11. Wykonać i zapisać zrzuty ekranu okna diagramu cykli zegarowych dla obu pętli.
Upewnić się że na zrzutach ekranu widoczne są nazwy i argumenty instrukcji.
12. Obliczyć CPI (Clock cycles Per Instruction) dla zoptymalizowanego kodu.

$$CPI = \frac{\text{liczba cykli zegarowych}}{\text{liczba wykonanych instrukcji}}$$

13. Utworzyć nowy dokument tekstowy. Dokument ma zawierać kolejno:
 - 13.1. Wartości zmiennych ns, np, g, n, m.
 - 13.2. Przydzielony wzór z [Tabeli 1.1](#).
 - 13.3. Przydzielony wzór z [Tabeli 1.2](#).
 - 13.4. Zrzut(y) ekranu okna pamięci operacyjnej przedstawiający wartości *A*.
 - 13.5. Zrzut(y) ekranu okna pamięci operacyjnej przedstawiający wartości *B*.
 - 13.6. Dwa zrzuty ekranu okna statystyk, z przed i po optymalizacji.
 - 13.7. Zawrzeć wykonane zrzuty ekranu okna diagramu cykli zegarowych w taki sposób, aby można było łatwo zaobserwować zmianę z przed i po optymalizacji,
 - 13.8. Obliczoną pisemnie liczbę cykli zegarowych, liczbę instrukcji i CPI.
14. Ustawić okna:
 - 14.1. Z lewej strony ekranu na drugim planie ma znajdować się kod programu.
 - 14.2. Z prawej strony ekranu na drugim planie ma znajdować się program WinDlx.

- 14.3. Z lewej strony ekranu na pierwszym planie ma znajdować się przygotowany dokument.
- 14.4. Z prawej strony ekranu na pierwszym planie ma znajdować się przygotowany arkusz kalkulacyjny.
- 15. Na blacie ma znajdować się kartka z obliczeniami.
- 16. Wpisać się na listę: <https://goo.gl/7Y17lh> i czekać na podejście prowadzącego.