**МАШИНИ З НАТУРАЛЬНОЗНАЧНИМИ РЕГІСТРАМИ:**

**Лекційний матеріал та завдання**

**(МАШИНИ З НЕОБМЕЖЕНИМИ РЕГІСТРАМИ)**

***Теорія для екзамену з предмету:***

Машина з натуральнозначними регiстрами (скорочено МНР) є iдеалiзованою моделлю комп’ютера. МНР мiстить нескiнченну кiлькiсть регiстрiв, вмiстом яких є натуральнi числа. Регiстри нумеруємо натуральними числами, починаючи з 1, позначаючи їх *R*0 *, R*1*, ..., Rn , ...* Вмiст регiстру *Rn* позначаємо *’Rn* .

Послiдовнiсть (*’R*0*, ’R*1*, ..., ’Rn , ...*) вмiстiв регiстрiв МНР назвемо *конфiгурацiєю* МНР.

МНР може змiнити вмiст регiстрiв згiдно виконуваної нею команди. Скiнченний список команд утворює *програму* МНР. Команди програми послiдовно нумеруємо натуральними числами, починаючи з 1. Номер команди в програмі називатимемо *адресою* команди. МНР-програму з командами *I*1*, I*2 *,..., Ik* будемо позначати *I*1*I*2*...Ik.* Довжину (кiлькiсть команд) МНР-програми *P* позначимо |*P*|.

Команди МНР бувають 4-х типiв.

*Тип* *1*. Обнулення *n*-го регiстру *Z(n): ’Rn* :=0*.*

*Тип* *2*. Збiльшення вмiсту *n-*го регiстру на 1 *S(n): ’Rn* :=*’Rn+*1*.*

*Тип* *3*. Копіювання вмісту регістру *T(m,n): ’Rn* :=*’Rm*

(при цьому *’Rm* не змiнюється).

*Тип 4.* Умовний перехiд *J(m,n,q):* *якщо ’Rn =’Rm , то перейти до виконання q-ї команди, iнакше виконувати наступну за списком команду програми*.

Число *q* в команді *J(m,n,q)* назвемо *адресою переходу*.

Команди типiв 1-3 називають *арифметичними*. Пiсля виконання арифметичної команди МНР повинна виконувати наступну за списком команду програми.

Виконання однiєї команди МНР назвемо *кроком* МНР.

Зауважимо, що формальними моделями алгоритмів є саме МНР-програми, поняття МНР використовується для опису функціонування МНР-програм.

Виконання програми МНР починає, перебуваючи в деякiй *початковiй* конфiгурацiї, з виконання 1-ї за списком команди. Наступна для виконання команда програми визначається так, як описано вище. Виконання програми завершується (програма зупиняється), якщо наступна для виконання команда вiдсутня (тобто номер наступної команди перевищує номер останньої команди програми). Конфiгурацiя МНР в момент завершення виконання програми називається *фiнальною*, вона визначає результат роботи МНР-програми над даною початковою конфiгурацiєю.

Якщо МНР-програма *P* при роботi над початковою конфiгурацiєю (*a*0*, a*1*, ...*) нiколи не зупиняється, цей факт позначаємо *P*(*a*0*, a*1*, ...*)↑, якщо ж коли-небудь зупиниться, цей факт позначаємо *P*(*a*0*, a*1*,...*)↓. Якщо МНР-програма *P* при роботi над початковою конфiгурацiєю (*a*0*, a*1*, ...*)зупиняється iз фiнальною конфiгурацiєю (*b*0*, b*1*, ...*), цей факт позначатимемо так: *P*(*a*0*, a*1*, ...*)↓(*b*0*, b*1*, ...*).

Функцiю *f:Nп*→*N* називають *МНР-обчислюваною*, якщо iснує МНР-програма, яка обчислює цю функцiю.

***Основні елементи практики:***

Стрічка МНР нескінченна в один бік. Кожна клітинка-регістр має номер, і за цим номером до неї можна звернутись в будь-який момент роботи.

…

Перший регістр

Четвертий регістр

В кожній клітинці-регістрі може бути тільки натуральне число або нуль. Якщо регістр порожній, то там знаходиться нуль.

Якщо задана функція , то в першому регістрі знаходиться значення змінної , в другому – , і третьому – .

4

1

7

…

Результат-відповідь на завдання після виконання програми завжди має бути в ПЕРШОМУ регістрі.

В МНР використовують тільки чотири команди – **обнулення *Z(n),* збільшення на 1 *S(n), к*опіювання вмісту регістру *T(m,n), у*мовний перехiд *J(m,n,q).***

|  |  |
| --- | --- |
| **Команда** | **Дія, яку реалізує МНР під час виконання команди** |
| *Z*(*n*) | *rn*:=0 |
| *S*(*n*) | *rn*:= *rn*+1 |
| *T*(*m*,*n*) | *rn*:= *rm* |
| *J*(*m*,*n*,*q*) | Якщо *rn*= *rm*, то перейти до виконання команди з номером *q*, у протилежному випадку виконувати наступну за порядком команду |

**Приклад на стрічці:**

Дана стрічка із наступними значеннями.

4

1

7

…

Після команди *Z(2)* буде (другий регістр заповниться нулем):

4

0

7

…

Після команди *S(3)* буде (значення третього регістру збільшиться на одиницю):

4

0

8

…

Після команди *T(3,1)* буде (те, що є в третьому регістрі скопіюється в перший, значення третього регістра не зміниться):

8

0

8

…

Вся програма МНР складається з даних команд. Всі команди програми нумеруються.

1. Z(2)
2. S(2)
3. T(2,1)

5

3

…

на початку роботи

5

0

…

після першої команди Z(2)

5

1

…

після другої команди S(2)

1

1

…

після третьої T(2,1)

Програма завершила свою роботу. Відповідь знаходиться у першому регістрі та дорівнює 1.

Програма зупиняється тоді, коли дойде до останньої команди, або відбудеться зациклення, або здійсниться перехід на номер відсутньої команди.

Після виконанної команди виконується наступна, за виключенням команди умовного переходу *J(m,n,q).*

**Команда *J(m,n,q)* :**

1. Спочатку порівнюються значення, які знаходяться в регістрах під номером *m,n*
2. Якщо значення-числа в регістрах *m,n* рівні, то здійснюється перехід на команду під номером *q*
3. Якщо значення в регістрах *m,n* не рівні, то здійснюється перехід на наступну по списку команду в програмі

*J(1,2,2)*

Це номери РЕГІСТРІВ, які порівнюються

Це номер КОМАНДИ, до якої слід перейти якщо значення клітинок рівні

1. Z(2)
2. S(2)
3. T(2,1)
4. J(1,2,2)

5

3

…

на початку роботи

5

0

…

після першої команди

5

1

…

після другої команди

1

1

…

після третьої

1

1

…

під час четвертої команди

порівнюються значення першого та другого регістрів (там стоять одиниці, тобто значення рівні), тоді здійснюємо умовний перехід на команду під номером 2.

п’ята команда – після переходу на другу команду виконується S(2)

1

2

…

шоста команда – далі по списку T(2,1)

2

2

…

сьома команда – далі по списку J(1,2,2)

2

2

…

два регістри (перший і другий рівні, тому переходимо на команду під номером 2)

Зрозуміло, що прогама зациклилась і відповідь не може бути записана.

**Переходимо до практичних завдань:**

1. *Записати програму МНР, що обчислює функцію*
2. Т(2,1) - значення *y* з другого регістру копіюється у перший регістр, де має бути результат
3. S(1) - значення першого регістру збільшується на 1
4. S(1) - значення першого регістру збільшується на 1
5. Z(2) - значення другого регістру очищується, онулюється
6. *Записати програму МНР, що обчислює функцію*
7. Z(1) - перший регістр онулюється
8. S(1) +1
9. S(1) +1
10. S(1) +1

Другий регістр, де *у*, можна онулити

1. *Записати програму МНР, яка завжди зациклена*
2. J(1,1,1)
3. *Записати програму МНР, що обчислює функцію*
4. J(1,2,4) - порівнюється *х* і *у*, якщо вони рівні, то в першій клітинці треба поставити одиницю (тому здійснюють перехід на 4 команду: першу клітинку обнулять, а потім додають 1)

Якщо клітинки не рівні, то виконується по списку наступна команда 2: T(3,1) третя клітинка порожня, там нуль, і даний нуль копіюємо в першу клітинку. Оскільки більше нічого не потрібно робити, тому наступна команда має зробити вихід з програми, перехід на команду, якої в програмі нема 6)

1. T(3,1)
2. J(1,1,6)
3. Z(1)
4. S(1)

Інший варіант

1. J(1,2,3)
2. J(2,2,4)
3. S(3)
4. T(3,1)

**Д/З: Розібратись з програмою:**

Показати, що наступна програма для МНР:

*І*1 *J*(1,2,3)

*І*2 *S*(2)

*І*3 *Т*(2,1)

обчислює функцію 

1. *Скласти програму для МНР, яка обчислює функцію f(x,у)=x+у.*

Розв’язок. На початку роботи програми стан регістрів МНР буде таким:

|  |  |  |  |
| --- | --- | --- | --- |
| *х* | *у* | 0 | 0 |
| *R*1 | *R*2 | *R*3 | *R*4 |

Ідея: До першої регістру *х* треба додати *у* одиниць, тобто стільки одиниць скільки їх є у другому регістрі. Тому виконують наступні дії: до першого і порожнього регістрів додають одиницю.

***S*(1)**

***S*(3)**

|  |  |  |  |
| --- | --- | --- | --- |
| *х* | *у* | 0 | 0 |
| *R*1 | *R*2 | *R*3 | *R*4 |
| *+1* |  | *+1* |  |

Цю дію требі повторити, тобто утворити цикл: *J*(1,1,1) – поверне до першої команди.

|  |  |  |  |
| --- | --- | --- | --- |
| *х* | *у* | 0 | 0 |
| *R*1 | *R*2 | *R*3 | *R*4 |
| *+1* |  | *+1* | Це треба зробити «**у»** раз |
| *+1* |  | *+1* |  |
|  | *…* |  |  |
| *+1* |  | *+1* |  |

Дії ***S*(1), *S*(3)** потрібно повторити «**у»** раз: тоді коли у третій клітинці теж буде «**у»**, тобто 2 і 3 регістри стануть рівними. Заготовка команди виходу з циклу *J*(2,3,\_\_). Якщо ці клітинки рівні, то ми додали до першого регістру «**у»** одиниць. Там буде х+у. Результат обчислено. І можна задати умовний перехід на неіснуючу команду 5. *J*(2,3,5)

Програма, що реалізує сформульовану задачу, виглядатиме так:

Умова виходу з циклу

*І*1 *J*(2,3,5)

*І*2 *S*(1)

Створення циклу

*І*3 *S*(3)

*І*4 *J*(1,1,1) .

**Пояснення на прикладі:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 2 | 0 | 0 | Нехай х=5, *у*=2. Тоді на 1-му такті при порівнянні вмісту регістрів *R*2 і *R*3 отримаємо не збігання (бо на початку роботи порівняли числа 2 і 0) |
| 6 | 2 | 0 | 0 | перейдемо до наступної команди, яка додасть у R1 одиницю та перейде до наступної команди. |
| 6 | 2 | 1 | 0 | На третьому такті роботи програми вміст регістра R3 збільшиться на 1 та здійсниться перехід до наступної команди |
| 6 | 2 | 1 | 0 | J(1,1,1). Значення першого регістра завжди з собою рівне, тому здійснюється перехід до першої команди даної програми. |
| 6 | 2 | 1 | 0 | На 5-му такті роботи програми команда J(2,3,5) порівнявши вміст регістрів R2 і R3 (тепер порівнюють числа 2 і 1) дасть можливість знову перейти до 2-ї та 3-ї команд запропонованої програми. |
| 7 | 2 | 2 | 0 | Виконання цих команд збільшить вміст регістрів R1 і R3 на 1. |
| 7 | 2 | 2 | 0 | На 8-му такті команда J(1,1,1) знову передасть керування першій команді. Оскільки тепер вміст регістрів R2 і R3 однаковий, то команда J(2,3,5) передасть керування на неіснуючий регістр R5 і програма завершить свою роботу. У регістрі R1 отримаємо результат виконання програми. |

1. *Скласти МНР-програму для функцiї f(x)=2x:*

*1.Перший спосіб: Скопіювати значення першого регістру у другий регістр і здійснити додавання чисел першого і другого регістру (х+х) згідно попереднього алгоритму*

***Д/З: Записати програму самостійно***

*2. Другий спосіб:У другий регістр додають одиницю, у третій дві одиниці (в два рази більше. Якщо умова була б 3х, то додавали б три одиниці).*

|  |  |  |  |
| --- | --- | --- | --- |
| *х* | *0* | 0 | 0 |
| *R*1 | *R*2 | *R*3 | *R*4 |
|  | *+1* | *+1*  *+1* |  |

Зациклюємо: J(1,1,1)

|  |  |  |  |
| --- | --- | --- | --- |
| *х* | *0* | 0 | 0 |
| *R*1 | *R*2 | *R*3 | *R*4 |
|  | *+1* | *+1*  *+1* |  |
|  | *+1* | *+1*  ***х*** ***2х***  *+1* |  |
|  | *…* |  |  |
|  | *+1* | *+1*  *+1* |  |

Зупинимо за ***х*** раз, тобто коли значення в першому і другому регістрах стануть рівними, тобто у другому регістрі буде ***х,*** у третьому ***2х***. Залишається тільки ***2х*** перемістити у перший регістр.

1. *J(1,2,6)*
2. *S(2)*
3. *S(3)*
4. *S(3)*
5. *J(1,1,1)*
6. *Т(3,1)*

**Д/З: Показати, що наступна програма для МНР:**

*І*1 *J*(1,2,6)

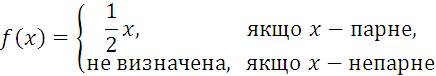
*І*2 *S*(3)

*І*3 *S*(2)

*І*4 *S*(2)

*І*5 *J*(1,1,1)

*І*6 *Т*(3,1)

обчислює функцію 

1. *Скласти МНР-програму для функцiї f(x,у)=x-у:*

Ідея:

|  |  |  |  |
| --- | --- | --- | --- |
| *х* | *у* | 0 | 0 |
| *R*1 | *R*2 | *R*3 | *R*4 |
|  | *+1* | *+1* |  |

«у» - другий регістр нарощуємо і третій регістр теж. Зациклюємо.

Зупиняємося тоді коли перший і другий регістр стануть рівними, тобто цикл буде здійснено стільки раз скільки одиниць не вистачало «у», щоб стати рівним «х».

1. *J(1,2,5)*
2. *S(2)*
3. *S(3)*
4. *J(1,1,1)*
5. *Т(3,1)*

**Д/З: Записати МНР-програми:**

1) *f*(*x*, *y*)=2*x*+*y*+1;

2) *f*(*x*)=(*x*+1)*/*3

3) *f*(*x*, *y*)=2*x*-*y*

*4*) *f*(*x*, *y, z*)=(*x**y*)+*z*;

**ДРУГА ЧАСТИНА ПРАКТИКИ:**

1. ***Записати програму МНР, що обчислює функцію***

*Ідея:* до вільного третього регістру додається одиниця, ще раз і ще раз. І порівнюють до якого значення ми швидше «дойдемо»: до *х* чи до *у. Якщо швидше доберемось до х, то воно менше, якщо до у, то воно менше.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 випадок | | | |  | 2 випадок | | | |
| *4* | *10* | 0 | 0 |  | 10 | 4 | 0 | 0 |
|  |  | +1 |  |  |  | +1 |  |
|  |  | +1 |  |  |  | +1 |  |
|  |  | +1 |  |  |  | +1 |  |
|  |  | +1 |  |  |  | +1 |  |
| *Ми швидше дойшли до значення 4 і воно є меншим і є відповіддю, значення знаходиться у першому регістрі* | | | |  | *Ми швидше дойшли до значення 4 і воно є меншим, залишилось його перемістити у перший регістр як відповідь* | | | |

*Розв’язання:*

1. *Спочатку записуємо ту дію яка буде повторюватись циклічно:*

*S(3) -команда циклу*

1. *Зациклюємо команду:*

*…*

*S(3)*

*J(1,1,\_\_)*

Номер команди для умовного переходу ще не прописуємо

1. Даний цикл має зупинись у двох випадках: або дойдемо до значення, що дорівнює х, або дойдемо до значення, що дорівнює у. Тобто буде два виходи з циклу

*J(3,1,\_\_) третій регістр швидше зрівнявся з першим, тобто там менше число це є відповіддю, достатньо зупинити роботу*

*J(3,2,\_\_) третій регістр зрівнявся швидше з другим, нам залишилось число з другого регістру перенести в перший*

*S(3)*

*J(1,1,\_\_)*

Підправляємо:

*J(3,1,\_\_)*

*J(3,2,\_\_)*

*S(3)*

*J(1,1,\_\_)*

*Т(2,1)*

Завершення роботи

1. Здійснимо нумерацію і заповнимо команди переходу:
2. *J(3,1, 6)*
3. *J(3,2, 5)*
4. *S(3)*
5. *J(1,1, 1)*
6. *Т(2,1)*

**Д/З Оформити знаходження максимуму двох чисел**

1. ***Записати МНР-програму для предикату "x=y":*** (завдання, що було в попередній лекції)

*Ідея: відповідь формуємо у третьому регістрі.*

*Розв’язання: порівнюємо два числа*

*J(1,2,\_)*

*Якщо числа рівні, то треба третій регістр збільшити на 1 і з третього регістру скопіювати значення в перший*

*J(1,2,\_)*

*S(3)*

*T(3,1)*

*Але якщо не рівні, то треба перенести нуль з третього регістру без збільшення на 1. Тобто потрібно побігти на команду T(3,1), оббігши команду S(3).*

*J(1,2,\_)*

*J(1,1,\_)*

*S(3)*

*T(3,1)*

*Команди нумеруємо і заповнюємо команди умовного переходу.*

1. *J(1,2,3)*
2. *J(1,1,4)*
3. *S(3)*
4. *T(3,1)*
5. ***Записати МНР-програму для предикату "x - парне":***

*Ідея: Ми будемо до другого регістру додавати 1 і намагатись «дойти» до заданого числа*

*J(1,2,\_)*

*S(2)*

*J(1,2,\_)*

*S(2)*

*Наприклад початкове значення х=5*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *J(1,2,\_)*  *S(2)*  *J(1,2,\_)*  *S(2)* | |  |  |  | | --- | --- | --- | | *5* | *0* | *порівняння* | | *5* | *1* | *приріст* | | *5* | *1* | *порівняння* | | *5* | *2* | *приріст* | | |  |  |  | | --- | --- | --- | | *5* | *2* | *порівняння* | | *5* | *3* | *приріст* | | *5* | *3* | *порівняння* | | *5* | *4* | *приріст* | | |  |  |  | | --- | --- | --- | | *5* | *4* | *порівняння* | | *5* | *5* | *приріст* | | *5* | *5* | *Порівняння. =*  *Необхідно здійснити перехід* | |  |  |  | |
| *Зациклення*  *J(1,1,\_)* | *Перше проходження циклу* | *Друге проходження циклу* | *Третє проходження циклу* |

*відповідь формуємо у третьому регістрі*

*J(1,2,\_)*

*S(2)*

Випадок парного числа

*J(1,2,\_)*

*S(2)*

*J(1,1,\_)*

Випадок непарного числа

*S(3)*

*T(3,1)*

*Зациклюємо і нумеруємо:*

1. *J(1,2,6)*
2. *S(2)*
3. *J(1,2,7)*
4. *S(2)*
5. *J(1,1,1)*
6. *S(3)*
7. *T(3,1)*

***Об’єднання двох попередніх задач:***

***Завдання:*** *Якщо число парне, то зменшити його вдвічі - поділити його на два, якщо ні – то залишити без змін.*

1. *J(1,2,6)*
2. *S(2)*

Якщо число парне, то переходимо до процесу ділення

1. *J(1,2,12)*
2. *S(2)*
3. *J(1,1,1)*
4. *J(1,4,11)*
5. *S(3)*
6. *S(4)*
7. *S(4)*
8. *J(1,1,6)*
9. *T(3,1)*
10. ***Записати програму МНР, що обчислює функцію***

*Ідея:*

*y раз*

*Тому виділяємо лічильник для обчислення скільки раз ми додали х - R*4.

*Результат додавання х буде записаний в R*5

***Розв’язання:***

*Формуємо цикл для додавання одного значення х*

*J(1,3,\_\_)*

*S(3) проміжний реєстр*

*S(5) реєстр результату*

*J(1,1,\_\_\_) зациклюємо*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Проміжне значення | Лічильник | Формування результату |
| *х* | *у* | 0 | 0 |  |
| *R*1 | *R*2 | *R*3 | *R*4 | *R*5 |
|  |  | *+1* |  | *+1* |

*Після циклу проміжне значення онулюємо Z(3)*

*Значення лічильника збільшуємо на 1 (до результату додали ще одне значення х) . S(4)*

*Зациклюємо, бо треба ще додати ще одне значення х. Тобто робимо зовнішній цикл, зупинимось тоді коли кількість х буде достатньою – коли їх кількість буде у: значення лічильника R*4 *буде дорівнює другого регістру де знаходиться у.*

*J(4,2,\_)*

*J(1,3,\_\_)*

Зовнішній цикл має відбутись «у» раз

Тобто стоп-кран циклу *J(4,2,\_)*

*S(3)*

*S(5)*

*J(1,1,\_\_\_)*

*Z(3)*

*S(4)*

*J(1,1,\_)*

*Нумеруємо та доповнюємо умовні переходи:*

1. *J(4,2,9)*
2. *J(1,3,6)*
3. *S(3)*
4. *S(5)*
5. *J(1,1,2)*
6. *Z(3)*
7. *S(4)*
8. *J(1,1,1)*
9. *Т(5,1)*
10. ***Записати програму МНР, що обчислює функцію***

*Задача зводиться до попередньої окремим випадком коли у=2.*

*1. S(2)*

*2. S(2)*

*3. …алгоритм множення*

***Д/З Записати програму МНР, що обчислює функцію 2х***

***Індивідуальні завдання:***

*Виконати завдання згідно порядкового номера в журналі*

|  |  |
| --- | --- |
|  | max(х-у,4z+5) |
|  | (х+у)/3+z/2 з перевіркою на ділення |
|  | min(у–х/3,x+y) |
|  | (у–3х)/z+2x з перевіркою на ділення |
|  | max(x,y/3)+z з перевіркою на ділення |
|  | (3х+3у)-x/2 +z з перевіркою на ділення |
|  | ((х+2у)/4)- z перевіркою на ділення |
|  | *3x(y+3)* |
|  | *(3x-2y-3)/3* з перевіркою на ділення |
|  | max(х-у,z+5) |
|  | (х+у)/2+z-1 з перевіркою на ділення |
|  | min(2x,x+y)-z |
|  | max(z–3х, х+2у) |
|  | *(x+y/2)-z+3x* |
|  | min(y-2,x+6-z) |
|  | ((х+у)/3)-2z перевіркою на ділення |
|  | max(z-у,4x+5) |
|  | min(у–х/3,z+y) |
|  | max(у–3х, х-3у) |
|  | max(x,y/3)+min(z,y) |
|  | (3х+3у)-z/2 перевіркою на ділення |
|  | max(у/2,4x+5) перевіркою на ділення |
|  | *(2x-3y)+z/5* перевіркою на ділення |
|  | ((х-2у)/3)+z/4 перевіркою на ділення |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Функція** | **Команда циклу** | **Зациклення** | **Команда виходу з циклу** | **На стрічці** |
| *x+y* | S(1)  S(3) | S(1)  S(3)  J(1,1,\_) | J(2,3,\_)  S(1)  S(3)  J(1,1,\_) | На початку роботи   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x | y |  |  |  |   В кінці роботи циклу   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x+y | y | y |  |  |   =  Умова виходу з циклу  J(2,3,\_) |
| *2x* | S(2)  S(3)  S(3) | S(2)  S(3)  S(3)  J(1,1,\_) | J(1,2,\_)  S(2)  S(3)  S(3)  J(1,1,\_) | На початку роботи   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x |  |  |  |  |   В кінці роботи циклу   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x | x | 2x |  |  |   =  Умова виходу з циклу  J(1,2,\_) |
| *x/2* | S(2)  S(3)  S(3) | S(2)  S(3)  S(3)  J(1,1,\_) | J(1,3,\_)  S(2)  S(3)  S(3)  J(1,1,\_) | На початку роботи   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x |  |  |  |  |   В кінці роботи циклу   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x | x/2 | x |  |  |   =  Умова виходу з циклу  J(1,3,\_) |
| x-y | S(2)  S(3) | S(2)  S(3)  J(1,1,\_) | J(1,2,\_)  S(2)  S(3)  J(1,1,\_) | На початку роботи   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x | y |  |  |  |   В кінці роботи   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x | x | x-y |  |  |   =  Умова виходу з циклу  J(1,3,\_) |