Задача 3. Даден е следният псевдо-код:

```
1   Program BestInterest
2   Interest, Base Rate, Balance: Real
3
4   Begin
5   Base Rate = 0.035
6   Interest = Base Rate
7
8   Read (Balance)
9   If Balance > 1000
10  Then
11      Interest = Interest + 0.005
12      If Balance < 10000
13      Then
14          Interest = Interest + 0.005
15      Else
16          Interest = Interest + 0.010
17      Endif
18  Endif
19
20      Balance = Balance * (1 + Interest)
21
22  End
```

а) Да се конструира граф на управляващия поток за псевдо-кода.

б) Да се опишат основните стъпки при конструиране на графа.

в) Да се дефинират тестови сценарии, като се използва конструираният граф, за да се получи пълно покритие на изразите (statements).

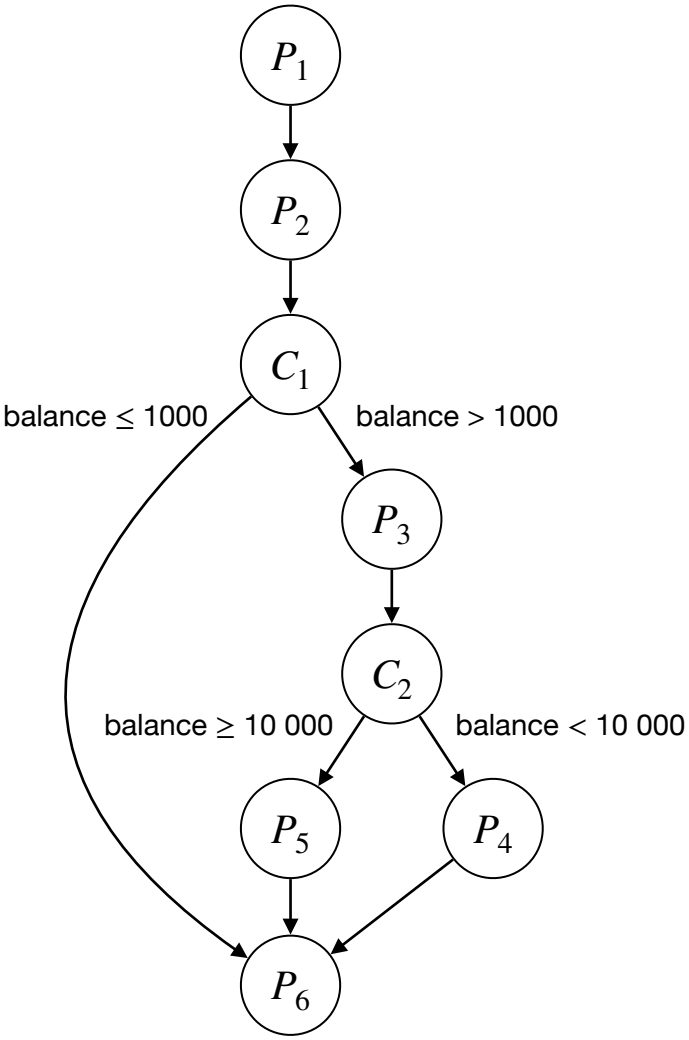
г) Даден е следният псевдо-код:

```
1   Begin
2   Read Time
3   If Time < 12 Then
4       Print(Time, "am")
5   Endif
6   If Time > 12 Then
7       Print(Time – 12, "pm")
8   Endif
9   If Time = 12 Then
10      Print(Time, "noon")
11  Endif
12  End
```

Да се определи какво ще бъде тестовото покритие на решенията (decisions) в проценти при тестови сценарии Time = 11 и Time = 15. Отговорът да се обоснове.

Решение.

а) Граф на управляващия поток за псевдо-кода.



P_1 : { Base Rate = 0.035; Interest = Base Rate

P_2 : Read(Balance)

C_1 : If Balance > 1000

P_3 : Interest += 0.005

C_2 : If Balance < 10 000

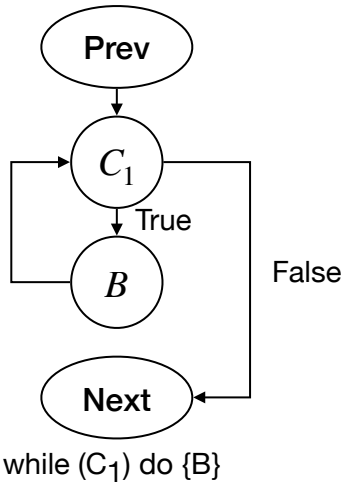
P_4 : Interest += 0.005

P_5 : Interest += 0.01

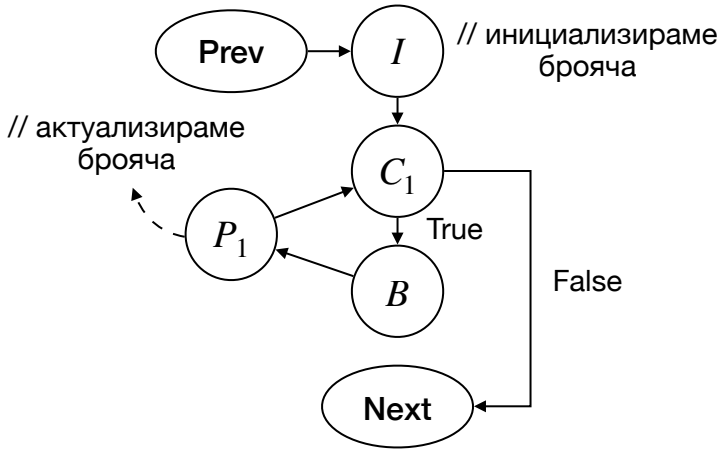
P_6 : Balance = (1 + Interest)

б) Основни стъпки при конструиране на графа на управляващия поток.

1. Асоцииране на обработващите възли P_1, P_2, ..., P_6 с изразите за присвояване, извикване на процедури или функции.
2. Асоцииране на възлите за взимане на решения C_1 и C_2 с изразите за условен преход: if-then-else или множествено разклонение "switch case".
3. Създаване на специален тип възли за разклонение и асоциирането им с изразите за цикъл



while (C_1) do {B}



for (I, C_1, P_1) do {B}

4. Асоцииране на началния и крайния възел на графа (P_1 и P_6) с първия и последния израз в програмата.

Ако в условието липвсаше псевдо код, стъпките щяха да бъдат:

1. Асоцииране на обработващите възли P_1, P_2, ..., P_6 с действия.
2. Асоцииране на разклоняващите възли C_1 и C_2 с условия и взимане на решения.
3. Асоцииране на началните и крайните възли (P_1 и P_6) съответно с първия и последния елемент в спецификацията.

При използването на тази техника за тестване освен конструиране на граф на управляващия поток, основните стъпки включват:

1. Конструитане и верифициране на графа на управляващия поток (на база блоксхеми, програмен код и документация).
2. Дефиниране и избор на пътища с цел покритие на определени тестови сценарии.
3. Определяне на входни стойности с цел изълнение на избраните пътища.
4. Изготвяне на план за проверка на резултата.

Забележка: Ако имаме цикли, всеки цикъл се тества със седем сценария:
1. bypass – **0** итерации
2. once – **1** итерация
3. twice – **2** итерации
4. typical – **max/2** итерации
5. **max-1** итерации
6. **max** итерации
7. **max+1** итерации

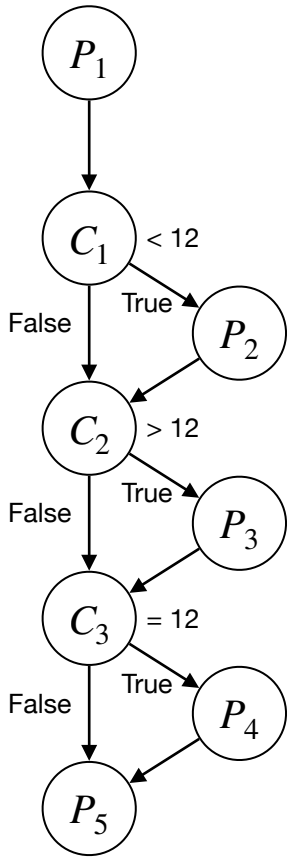
в) Дефиниране на тестови сценарии с цел пълно покритие

1. balance = 5000 за $C_1 =$ True и $C_2 =$ True;
2. balance = 15 000 за $C_1 =$ True и $C_2 =$ True;

Чрез тези тестови сценарии ще получим пълно покритие поради пътищата, които съществуват в нашия граф.

За да получим по-добро изследване на поведението на системата, може да комбинираме тази техника на тестване с „анализ на графичните стойности“ и тогава допускайки, че системата работи с точност до 2 цифри след десетичната запетая, мопже да дефинираме и допълнителни тестови сценарии за balance = 1 001.00 и balance = 9 999.99.

г) Какво ще бъде тестовото покритие в % при Time = 11 и Time = 15



Първият тестов сценарии покрива $C_1 = \text{True}$, $C_2 = \text{False}$, $C_3 = \text{False}$. Вторият тестов сценарии покрива $C_1 = \text{False}$, $C_2 = \text{True}$, $C_3 = \text{False}$.

Единствено не е покрит случая, при който $C_3 = \text{True}$, а всички са $C_1 = \text{True/False}$, $C_2 = \text{True/False}$, $C_3 = \text{True/False}$, т.е. 6, тъй като проверките C_1, C_2 и C_3 се осъществяват последователно в кода.

Изчисляването на третият условен израз като True не е покрит от тестовите сценарии.

Отговор: Тестовото покритие на решенията е 83% (5/6)

Основни стъпки при тестване на домейна:
1. Идентифициране на входната променлива, вектор, пространство и дефиниране на входния домейн;
2. Разделяне на входния домейн на поддомейни;
3. Анализ на поддомейните с цел определяне на границите им по всички измерения;
4. Избор на тестови точки (сценарии), покриващи поддомейните;
5. Тестване с избраните тестови точки, проверка на резултатите, решаване на проблеми и извършване на анализи.

Тестване с комбинация на екстремни точки

Асоциира се множество от тестови точки с всеки поддомейн. Всяка променлива X_i се намира:

– min_i – минимална стойност на X_i;

– max_i – максимална стойност на X_i;

– $under_i$ – стойност, малко по-голяма от min_i;

– $over_i$ – стойност, малко по-голяма от max_i;

– $interior$ – вътрешна стойност за поддомейна.

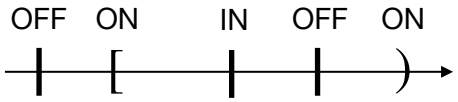
$\Rightarrow$ Брой тестови сценарии при n променливи: $4^n + 1$. За $0 \leq X < 21 \longrightarrow$ тестови точки -1, 0, 10, 20, 21

Стратегия weak N x 1

– Избор на 1 „OFF“ точка (точка, която НЕ е на границата)
– Избор на n „ON“ точки (точки, които са на границата)

– При отворена граница „ON“ точките се обработват като външни.
  – Като „OFF“ точка се избира вътрешна точка, близка до границата
– При затворена граница „ON“ точките се обработват като вътрешни
  – Като „OFF“ точка се избира външна точка, близка до границата

Брой тестови сценарии при b граници: $(n+1) \times b + 1$ за $0 \leq X < 21$ – тестовите точки са:



Стратегия weak 1 x 1

- Само една „ON“ точка за всяка граница
  - Броят на тестовите сценарии при b на брой граници на поддомейните е $2b + 1$.
- „OFF“ точката е на дистанция ε от „ON“ точката, перпендикулярна на границата.

Тестване на опашки

- Долна граница при 0, 1 или 2 елемента в опашката;
- Горна граница при $B - 1, B, B + 1$ елемента, където B е максималния брой елементи в опашката;
- Тестов сценарии за тестване на поведението при нормални обстоятелства ($B/2$ елемента).

□