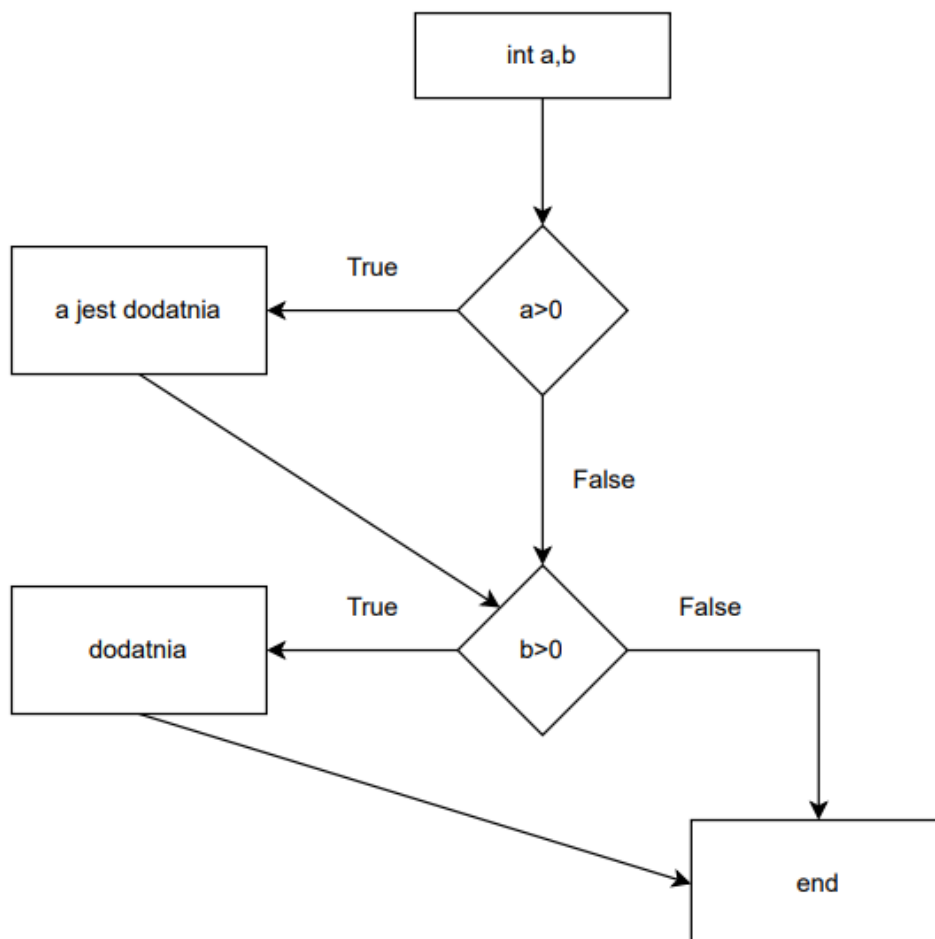


PRZYKŁAD 1**Pseudokod:**

```
int a,b
if (a>0) {
    printf („a jest dodatnia”)
}
if (b>0) {
    printf („dodatnia”)
}
end
```

Schemat blokowy:

Przypadki testowe:

Id	Przypadek testowy	Opis przypadku	Kroki testu	Opis kroku	Oczekiwany rezultat	Aktualny rezultat	Status	Komentarze
1	Prawidłowe identyfikowanie wprowadzonych wartości "a" i "b".	Użytkownik ma możliwość odczytać czy wprowadzone wartości "a" i "b" są dodatnie.	Warunek wstępny	Uruchomienie aplikacji. Przygotowanie danych testowych (cyfry większe od 0).				
			Krok 1	Wprowadzenie do aplikacji wartości "a" większej niż 0.	Aplikacja zwraca komunikat: "a jest dodatnia", a następnie kończy swoje działanie.	Aplikacja zwraca komunikat: "a jest dodatnia", a następnie kończy swoje działanie.	Pass	
			Krok 2	Wprowadzenie do aplikacji wartości "b" większej niż 0.	Aplikacja zwraca komunikat: "dodatnia", a następnie kończy swoje działanie.	Aplikacja zwraca komunikat: "dodatnia", a następnie kończy swoje działanie.	Pass	
			Krok 3	Usunięcie wprowadzonych danych.	Dane są możliwe do usunięcia. System jest gotowy do wprowadzenia nowych wartości "a" i "b".	Dane są możliwe do usunięcia. System jest gotowy do wprowadzenia nowych wartości "a" i "b".	Pass	

PRZYKŁAD 2

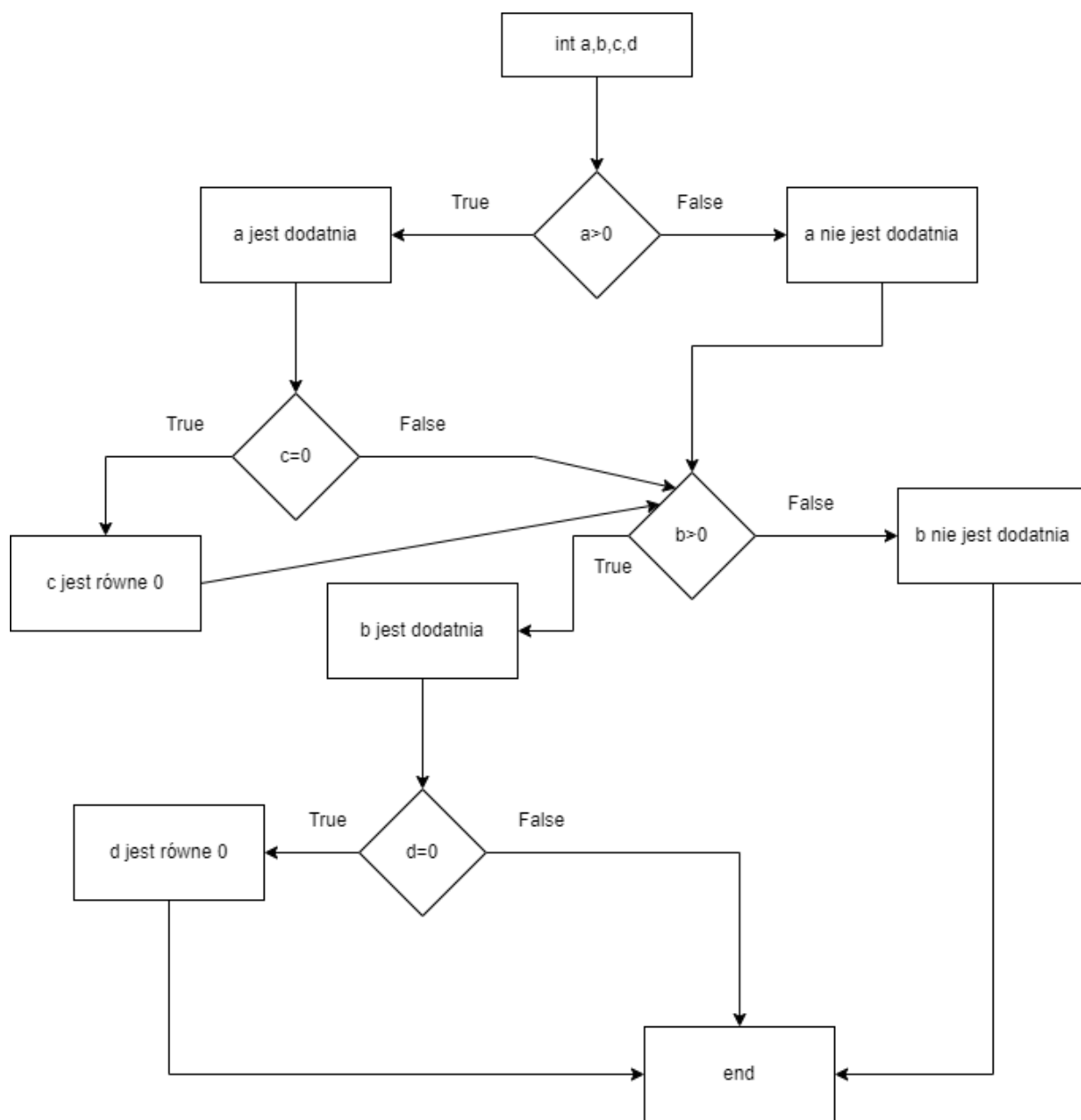
Pseudokod:

```

int a,b,c,d
if (a>0) {
    printf („a jest dodatnia”)
    if (c==0) {
        printf („c jest równe 0”)
    }
} else {
    printf („a nie jest dodatnia”)
}
if (b>0) {
    printf („b jest dodatnia”)
    if (d==0) {
        printf („d jest równe 0”)
    }
} else {
    printf („b nie jest dodatnia”)
}
end

```

Schemat blokowy:



Przypadki testowe:

Id	Przypadek testowy	Opis przypadku	Kroki testu	Opis kroku	Oczekiwany rezultat	Aktualny rezultat	Status	Komentarze
1	Prawidłowe identyfikowanie wprowadzonych wartości "a" i "b" większych od 0.	Użytkownik ma możliwość odczytać czy wprowadzone wartości "a" i "b" są dodatnie, a "c" i "d" równe 0.	Warunek wstępny	Uruchomienie aplikacji. Przygotowanie danych testowych (cyfry większe od 0).				
				Krok 1 Wprowadzenie do aplikacji wartości "a" większej niż 0.	Aplikacja zwraca komunikat: "a jest dodatnia".	Aplikacja zwraca komunikat: "a jest dodatnia".	Pass	
				Krok 2 Wprowadzenie do aplikacji wartości "c" równej 0.	Aplikacja zwraca komunikat: "c jest równa zero".	Aplikacja zwraca komunikat: "c jest równa zero".	Pass	
				Krok 3 Wprowadzenie do aplikacji wartości "b" większej niż 0.	Aplikacja zwraca komunikat: "b jest dodatnia".	Aplikacja zwraca komunikat: "b jest dodatnia".	Pass	
				Krok 4 Wprowadzenie do aplikacji wartości "d" równej 0.	Aplikacja zwraca komunikat: "d jest równa zero", a następnie kończy działanie.	Aplikacja zwraca komunikat: "d jest równa zero", a następnie kończy działanie.	Pass	
2	Prawidłowe identyfikowanie wprowadzonych wartości "a" i "b" mniejszych od 0.	Użytkownik ma możliwość odczytać czy wprowadzone wartości "a" i "b" nie są dodatnie.	Warunek wstępny	Usunięcie wprowadzonych danych.	Dane są możliwe do usunięcia. System jest gotowy do wprowadzenia nowych wartości "a", "b", "c" i "d".	Dane są możliwe do usunięcia. System jest gotowy do wprowadzenia nowych wartości "a", "b", "c" i "d".	Pass	
				Krok 1 Wprowadzenie do aplikacji wartości "a" mniejszej niż 0.	Aplikacja zwraca komunikat: "a nie jest dodatnia".	Aplikacja zwraca komunikat: "a nie jest dodatnia".	Pass	
				Krok 2 Wprowadzenie do aplikacji wartości "b" mniejszej niż 0.	Aplikacja zwraca komunikat: "b nie jest dodatnia", a następnie kończy działanie.	Aplikacja zwraca komunikat: "b nie jest dodatnia".	Pass	
				Krok 3 Usunięcie wprowadzonych danych.	Dane są możliwe do usunięcia. System jest gotowy do wprowadzenia nowych wartości "a", "b", "c" i "d".	Dane są możliwe do usunięcia. System jest gotowy do wprowadzenia nowych wartości "a", "b", "c" i "d".	Pass	

PRZYKŁAD 3

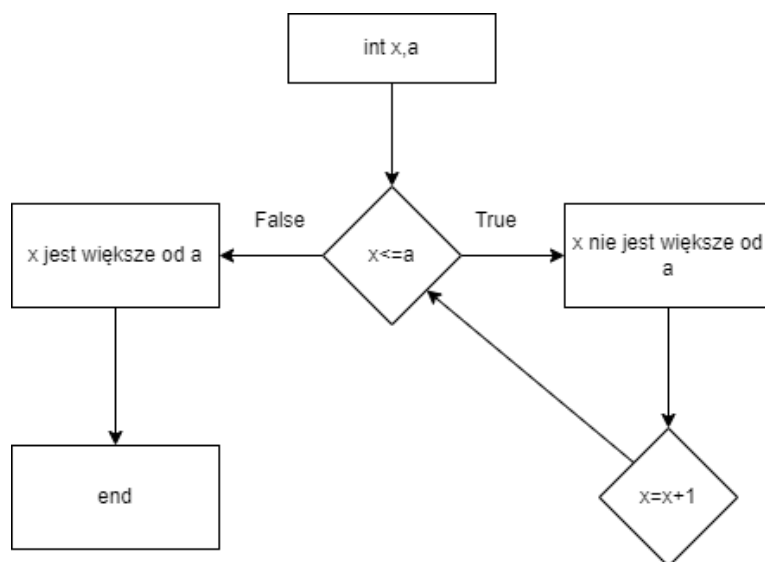
Pseudokod:

```

int x,a
while (x<=a) {
    printf („x nie jest większe od a”)
    x = x+1
}
printf („x jest większe od a”)
end

```

Schemat blokowy:



Przypadki testowe:

Przypadek Id	testowy	Opis przypadku	Kroki testu	Opis kroku	Oczekiwany rezultat	Aktualny rezultat	Status	Komentarze
1	Prawidłowe zwiększanie wartości "x"	Użytkownik ma możliwość automatycznie zwiększyć wartość "x" do momentu osiągnięcia wartości większej niż wskazane "a".	Warunek wstępny	Uruchomienie aplikacji. Przygotowanie danych testowych ("x" mniejsze bądź równe "a").				
				Krok 1 Wprowadzenie do aplikacji wartości "a".	Aplikacja umożliwia wprowadzenie wartości "a".	Aplikacja umożliwia wprowadzenie wartości "a".	Pass	

Krok 2	Wprowadzenie do aplikacji wartości "x" mniejszej bądź równej "a".	Aplikacja zwraca komunikat: "x nie jest większe od a". Dodaje do wartości "x" + 1 do momentu osiągnięcia wartości wyższej, niż wskazane "a", a następnie kończy działanie.	Aplikacja zwraca komunikat: "x nie jest większe od a". Dodaje do wartości "x" + 1 do momentu osiągnięcia wartości wyższej, niż wskazane "a", a następnie kończy działanie.	Pass	
	Usunięcie wprowadzonych danych.	Dane są możliwe do usunięcia. System jest gotowy do wprowadzenia nowych wartości "a", "x".	Dane są możliwe do usunięcia. System jest gotowy do wprowadzenia nowych wartości "a", "x".	Pass	
Krok 3					