

Algorytmy i struktury danych I

Zestaw zadań nr. 1: Poprawność semantyczna

• Zadanie 1

Sprawdź poprawność specyfikacji:

```
/* x >= 0 */ (warunek wstępny)
x = x+1
/* x >= 1 */ (warunek końcowy)
```

• Zadanie 2

Sprawdź poprawność specyfikacji dla sekwencyjnego wykonania instrukcji

```
/* x >= 1 */ (warunek wstępny)
y = x
/* y >= 1 */ (warunek końcowy)
```

```
/* y >= 1 */ (warunek wstępny)
y = y+1
/* y >= 2 */ (warunek końcowy)
```

• Zadanie 3

Udowodnij poprawność warunku końcowego dla następującego kodu.

```
/**/ (warunek wstępny)
if( x>=0 ) y=y+x
else y = y-x
/* y = y0 + |x0| */(warunek końcowy)
```

• Zadanie 4

Podaj niezmiennik dla poniższej pętli i udowodnij jego poprawność:

```
/* 1 <= b */ (warunek wstępny)
i=1; c=1;
while ( i<= b) { c = c * a ; i++};
```

• Zadanie 5

Udowodnij poprawność warunku końcowego dla następującego kodu:

```
/*x >= 100 */
if( x >= y )
    m = x;
else
    m = y;
/* m >= 100 */
```

- Zadanie 6

Napisz program, który wczytuje ciąg liczb dodatnich, zakończony przez -1, oblicza i wypisuje największą z wczytanych liczb dodatnich. Podaj niezmiennik dla pętli w Twoim programie, a następnie udowodnij, że jest on poprawny i że wynika z niego iż program robi to co powinien.

- Zadanie 7

Napisz program który wczytuje ciąg 100 liczb dodatnich, oblicza i wypisuje największą z wczytanych liczb dodatnich. Podaj niezmiennik pętli w Twoim programie, a następnie udowodnij, że jest on poprawny i że wynika z niego, iż program robi to co powinien.