Podprogramy. Przekazywanie parametrów.

1. 1. Przeanalizuj program historia.c. Usuń modyfikator static w deklaracji zmiennej suma w funkcji powiększaj. Czy wartości zwracane przez kolejne wywołania funkcji się zmieniły? Tak. Ze static wartości to 1 4 9, a bez to 1 3 5.
2. Dlaczego? Modyfikator static umożliwia pamiętanie wartości zmiennej po wykonaniu funkcji co za tym idzie przy następnym wywołaniu nie jest przypisywana nowa wartość do zmiennej tylko wykorzystywana obecnie przypisana. W zwykłych zmiennych wartość zmiennej po wykonaniu funkcji jest zapominana.
3. 2. Przeanalizuj/wykonaj/poeksperymentuj z programem wskaźniki.c
4. 3. Przeanalizuj program przekazTablice.c. W jaki sposób przekazywane są parametry aktualne do podprogramów generuj i pokaz (przez wartość/przez wynik/przez wynik i wartość/przez referencję).
5. Myślę że przez referencje ponieważ nigdzie niema odwołania do miejsca w pamięci zmiennych.
   1. Uzupełnij program przekazTablice.c o instrukcje, które udowodnią Twoją tezę. Uzupełniony program wklej   
      tutaj:
6. Przeanalizuj program *zamiana.c.* 
   1. Odgadnij jakie będą wartości zmiennych zm oraz elementów tablicy lista po wykonaniu programu ………………2 i 2…………………….. Zweryfikuj swoją diagnozę wykonując program.   
      Czy po wykonaniu funkcji zamien zmienia się zawartość tablicy lista …………………………..  
      Jaki sposób przekazywania parametrów ((przez wartość/przez wynik/przez wynik i wartość/przez referencję) został tu użyty. .……………………………………...
   2. Uruchom program zamiana.adb, uzupełnij program główny o wyświetlenie tablicy lista. Czy program działa tak samo jak zamiana.c? …………………………………………………………………………………….. Dlaczego? ------------------
   3. Zbadaj skutki zmiany sposobu przekazywania parametrów w programie zamiana.adb na IN; na OUT ……………………………………………………………………..   
      Wyjaśnij dlaczego tak się dzieje ………………………
   4. Zmień definicję funkcji zamien w programie zamiana.c , tak, aby zamiana tam dokonywana była widoczna w programie głównym, tak jak w programie zamiana.adb. (wskazówka: parametry przekazuj przez referencję; możesz przepisać program na c++ )
7. Uruchom, przeanalizuj program *zwrocTablice.adb (*kompilacja*:* gnatmake *zwrocTablice)*
   1. W jaki sposób przekazywany jest parametr do funkcji Odwroc\_Wektor (przez wartość/przez wynik/przez wynik i wartość/przez referencję/przez nazwę). …………………………………………………………..
   2. Wykonaj polecenia podane w komentarzach do programu i sprawdź działanie programu po poprawkach. Uzupełniony program wklej tu:
   3. Jaka struktura jest zwracana przez funkcję Odwroc\_Wektor …………………………………………..
   4. Napisz w C/C++ program analogiczny do zwrocTablice. Czy możesz zastosować „chwyty” zastosowane w Adzie? Program w C wklej tu:
   5. W jaki sposób „uchronić” tablicę przekazywaną jako parametr do funkcji (w C) odwroc\_Wektor przed zmianą wartości? Propozycję programu wklej tu:
8. Przeanalizuj program *funArg.c oraz funArg.cpp*. Ten program podpowie Ci jak napisać funkcję *calkowanie*, która dla podanych granic i podanej funkcji obliczy całkę oznaczoną tej funkcji w tych granicach.   
   Program na całkowanie wklej tu:
9. Napisz program w języku, w którym można stosować parametry w „czystym” trybie wyjściowym, tzn. out, a nie in-out (np. Ada). Program powinien pokazać, w którym momencie ów parametr jest wiązany z konkretną komórką pamięci — czy w chwili wywołania podprogramu, czy w chwili jego zakończenia, czy może jeszcze kiedy indziej.