DOKUMENTACJA

1.Opis problemu i rozwiązania

Naszym zadaniem jest reprezentacja działań na wielomianach w programie za pomocą tablic. Operacje jakie musimy zaimplementować to dodawanie, odejmowanie, mnożenie oraz obliczanie wartości wielomianu w punkcie za pomocą algorytmu Hornera. W programie będziemy przechowywać tylko wartości współczynników wielomianów oraz stopień. Współczynniki będziemy przechowywać w tablicy, gdzie wiel[i] oznacza współczynnik przy x^i.
Pierwszym etapem rozwiązania problemu jest wczytanie wielomianu od użytkownika. Użytkownik podaje stopień wielomianu, a następnie współczynniki od najwyższej potęgi do stałej. Aby uniknąć błędów, program sprawdza poprawność wprowadzonych danych, zapewniając, że stopień jest liczbą całkowitą nieujemną.

```
void wczytajWielomian(double wiel[], int stopien)
  wypisz: „Podaj stopień wielomianu:”
  wczytaj: stopien
 if(stopien<0) wypisz: „Podano błędną liczbę.”
 wypisz: „Podaj współczynniki (od najwyższej potęgi do stałej):”
  for( i = stopien; i>=0;i--) wczytaj: wiel[i]
```

Po wczytaniu wielomianów możemy przejść do ich przetwarzania. Operacja dodawania polega na sumowaniu współczynników odpowiadających tym samym potęgom. Stopień wyniku jest równy maksymalnemu stopniowi z obu wielomianów. Podobnie wygląda odejmowanie.

```
void dodaj(wiel1[], stopien1, wiel2[], stopien2, wynik[], stopienWyniku)
  stopienWyniku = MAX(stopien1, stopien2)
   for( i = 0 ; i<= stopienWyniku; i++)
     wynik[i] = 0
     if( i ≤ stopien1)  wynik[i] += wiel1[i]
     if (i ≤ stopien2)  wynik[i] += wiel2[i]


void odejmij(wiel1[], stopien1, wiel2[], stopien2, wynik[], stopienWyniku)
  stopienWyniku = MAX(stopien1, stopien2)
  for( i = 0; i <= stopienWyniku; i++)
     wynik[i] = 0
     if (i ≤ stopien1)  wynik[i] += wiel1[i]
     if (i ≤ stopien2)  wynik[i] -= wiel2[i]
```

Następną operacją jest mnożenie wielomianów. W tym przypadku każdy współczynnik pierwszego wielomianu należy pomnożyć przez każdy współczynnik drugiego wielomianu, przy czym potęgi zmiennej x sumują się. Stopień wyniku mnożenia będzie równy sumie stopni obu wielomianów.

```
void mnozenie(wiel1[], stopien1, wiel2[], stopien2, wynik[], stopienWyniku)
```

```
stopienWyniku = stopien1 + stopien2

for(i = 0 ; i<= stopienWyniku; i++)
   wynik[i] = 0

for(i = 0 ; i<= stopienWyniku; i++)
    for(j = 0 ;j<= stopienWyniku; j++)
       wynik[i + j] += wiel1[i] * wiel2[j]
```

Ostatnią operacją, jaką musimy zaimplementować, jest obliczanie wartości wielomianu dla podanego argumentu x. W tym celu stosujemy algorytm Hornera, który znacząco upraszcza i przyspiesza obliczenia.

```
double Horner(wiel[], stopien, x)
 wynik = wiel[stopien]

 for(i=stopien-1; i>=0;i--)
   wynik = wynik * x + wiel[i]

 return wynik
```

2. Opis użytych struktur danych

Wielomiany są reprezentowane jako tablice typu double, w których każda komórka zawiera współczynnik odpowiadający odpowiedniej potędze zmiennej x. Tablice zostały zadeklarowane z maksymalnym rozmiarem 100, co oznacza, że wielomian może być co najwyżej 99 stopnia. W kodzie wykorzystywane są także zmienne całkowite int, które przechowują stopień odpowiedniego wielomianu. Zmienne te pozwalają programowi zarządzać strukturą wielomianów i kontrolować zakres obliczeń. Wartością zmienną w tym programie jest również zmienna x typu double, która przechowuje wartość punktu, w którym obliczana jest wartość wielomianu za pomocą algorytmu Hornera.

3. Oszacowanie złożoności czasowej i pamięciowej

- Wczytywanie i wyświetlanie wielomianów
  Pętla iteruje przez n+1 elementów, dlatego wczytywanie współczynników i wyświetlanie wielomianu to operacje liniowe O(n), gdzie n to stopień wielomianu. Nie używają dodatkowej pamięci poza przechowywaniem współczynników, dlatego złożoność pamięciowa to O(n).

- Algorytm Hornera
  Oblicza wartość wielomianu w punkcie x poprzez kolejne mnożenia i dodawania współczynników. W każdej iteracji wykonujemy jedną operację mnożenia i jedną operację dodawania – O(1) ale pętla wykonuje się n razy, co wymaga O(n) operacji dla wielomianu stopnia n. Algorytm nie wymaga dodatkowej pamięci poza danymi wejściowymi, stąd jego złożoność pamięciowa wynosi O(1).

- Dodawanie i odejmowanie wielomianów
  Dodajemy lub odejmujemy odpowiednie współczynniki, tyle razy ile wynosi maksymalny stopień spośród dwóch wielomianów, n=max(stopien1,stopien2). Musimy wykonać co najwyżej n+1 operacji dodawania lub odejmowania więc złożoność obliczeniowa to O(n). Złożoność pamięciowa również wynosi O(n), ponieważ używamy dodatkowej tablicy o rozmiarze n+1 do przechowania wyniku.

- Mnożenie wielomianów
  Pętla zewnętrzna iteruje przez współczynniki pierwszego wielomianu - O(n), a wewnętrzna przez współczynniki drugiego - O(m), (n>=m) co prowadzi do O(n·m) operacji. W najgorszym przypadku n=m, więc mamy O(n^2). Wynikowy wielomian ma stopień = stopien1+stopien2, więc tablica wynikowa wymaga przechowywania n+m+1 elementów co daje jej złożoność pamięciową O(n+m) .

4. Dokumentacja użytkowa

Program jest napisany w języku C++ i może być uruchomiony na dowolnym systemie operacyjnym, który obsługuje kompilator C++.
Najpierw należy program pobrać i skompilować. Dla linuxa wystarczy wpisać w terminalu: g++ algorytmy.cpp -o algorytmy
a następnie uruchomić wpisując: ./algorytmy
Po uruchomieniu programu użytkownik zobaczy menu, w którym będzie musiał wybrać jedną z czterech dostępnych opcji:

1 dodawanie

2 odejmowanie

3 mnożenie

4 obliczanie wartości wielomianu w punkcie

Użytkownik musi wpisać wybraną cyfrę i nacisnąć enter. Następnie program poprosi o wpisanie stopnia wielomianu, tutaj użytkownik musi podać cyfrę z przedziału od 0 do 99 po czym nacisnąć enter. Kolejno program poprosi o podanie współczynników wielomianu, użytkownik musi wpisać je zaczynając od współczynnika stojącego przy najwyższej potędze do wyrazu wolnego, oddzielając je spacją i na końcu wcisnąć enter. Współczynniki mogą być liczbami zmiennoprzecinkowymi, wtedy należy wpisać część_całkowita.część_ułamkowa. W przypadku wybrania opcji 1,2,3 program znowu poprosi o wpisanie stopnia i współczynników a dla opcji 4 program poprosi o wpisanie punktu, który również może być liczbą zmiennoprzecinkową i wciśnięcie enter. Jeśli którakolwiek z danych została podana błędnie, program poprosi o ponowne jej wprowadzenie.