Przedmiot: Wprowadzenie do programowania

Dokumentacja projektowa (techniczna) i użytkowa

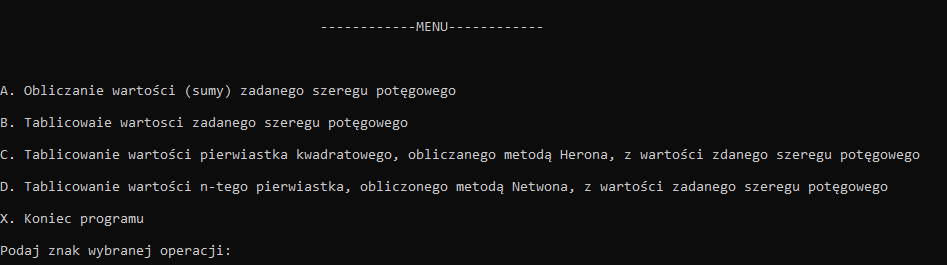
projektu numer 2

Antoni Nowalski 57295

Treść zadania projektowego

Zaprojektuj (w języku CSharp) program: *Obliczenia Iteracyjne,* którybędzie umożliwiał:

- wielokrotną realizację (obliczanie) poszczególnych funkcjonalności programu,

- wybór odpowiednej funkcjonalności (wynikającej z jego potrzeb obliczeniowych) przez naciśnięcie jej klawisza:

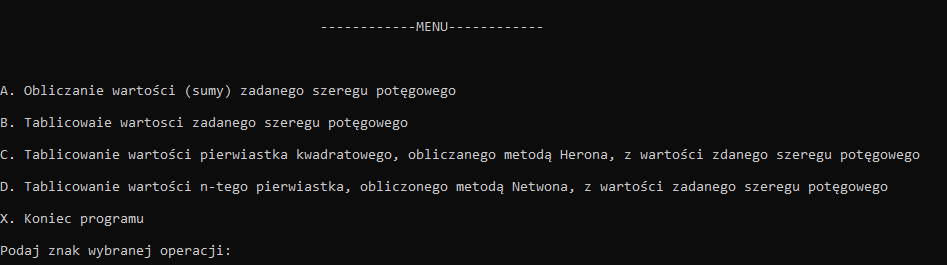
-realizację wybranej funkcjonalności programu w dwóch trybach:

-*bez śledzenia*, czyli z bezpośrednim wypisaniem wyniku obliczeń,

-*ze śledzeniem*, czyli z monitorowaniem (śledzeniem) kolejnych kroków obliczeń.

W implementacji (rozwiązaniu) *Projektu Nr 2* zaprojektuj metody dla obsługi funkcjonalności programu (do rozpatrywania wybranej funkcjonalności zastosuj instrukcję switch)

1.Obliczenie wartości (sumy) zadanego ciągu potęgowego za pomocą metody: anSumaSzereguPotegowego



a)Krótki opis funkcjonalności:

-Oblicza wartość zadanego ciągu potęgowego z podanych przez użytkownika wartości z użyciem dodatkowych metod: anwczytaniedouble, anwczytanieEps.(Zostaną one krótko opisane pod koniec dokumentacji).

b)Dane wejściowe wraz z typami:

-double anx, aneps.

-bool anczysledzenie.

-ushort anlicznik.

c)Kolejność wyczytywania danych:

-anx, aneps, anczysledzenie, aneps.

d)Warunki wprowadzanych danych:

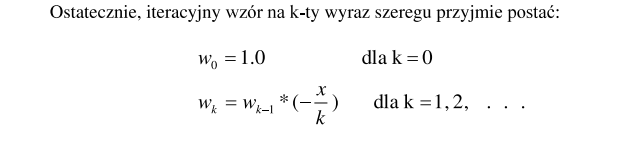
- anx: liczba typu double,

- aneps: liczba z zakresu (0;1).

e)Postać wypisywania wyników:

-pojedyncza wartość.

f)Wzór:



g)Kod C#

static double anSumaSzereguPotegowego(double anx, double aneps, out ushort anlicznik, bool anczysledzenie)

{

double anw = 1, ansuma = 1;

anlicznik = 1;

while (Math.Abs(anw) > aneps)

{

anlicznik++;

anw \*=(-anx / anlicznik);

ansuma += anw;

if(anczysledzenie)

Console.WriteLine("\n\n\tŚledzenie: W {0} -tej iteracji: SumaSzeregu = {1, 5:F2} , a wartość wyrazu W = {2, 5:F2}", anlicznik-1, ansuma, anw);

}

return ansuma;

}

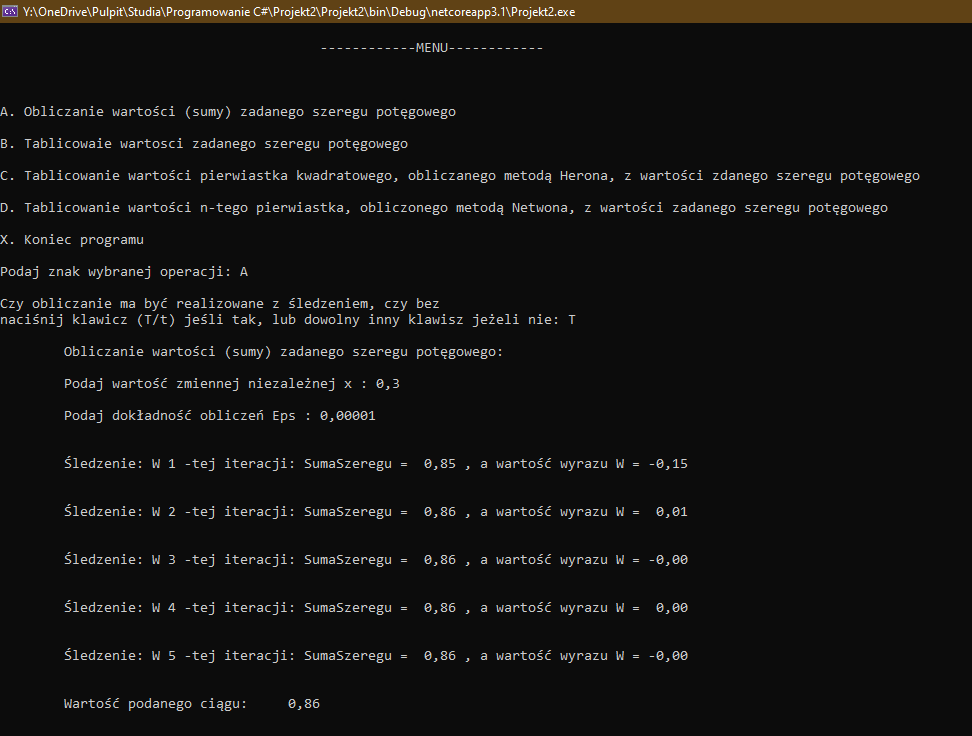
h)Dokumentacja testowania funkcjonalności:

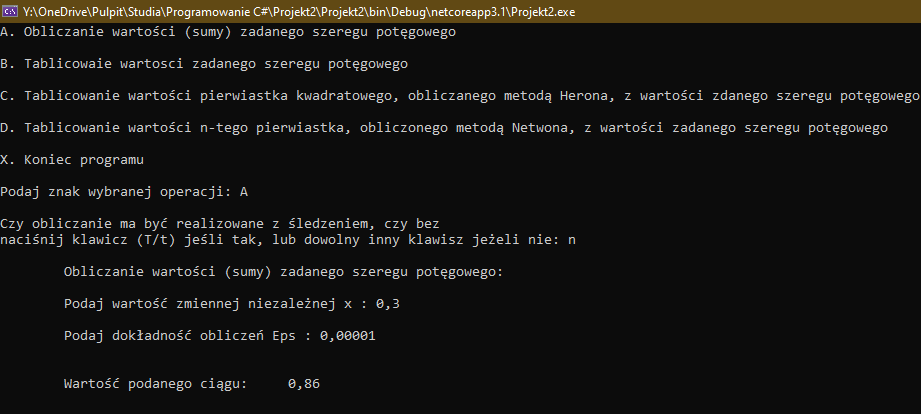
-Dane: anx = 0.3, aneps = 0.00001

-Wynik: 0.86

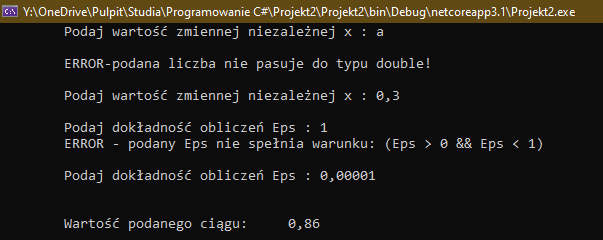
-Screen:

Z śledzeniem:

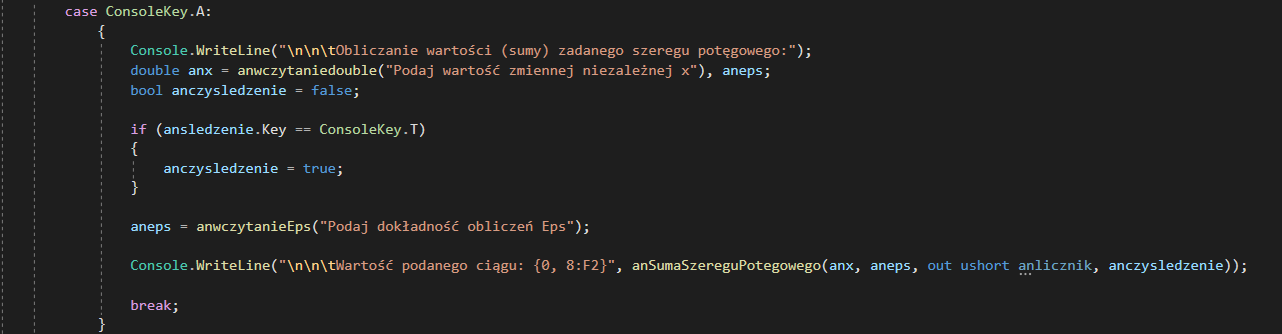


Bez śledzenia:

Błędne dane:



i)W main:



2. Tablicowanie wartości zadanego szeregu potęgowego za pomocą metody: anTablicowanieZmianyWartosciSzereguPotegowego.

a)Krótki opis funkcjonalności:

-Wyświetlenie wartości zadanego szeregu potęgowego w formie tablicy z użyciem dodatkowych metod: anSumaSzereguPotegowego, anSkok, anPrzedzial, anwczytanieEps(Zostaną one krótko opisane pod koniec dokumentacji).

b)Dane wejściowe wraz z typami:

-double: anpocz, ankon, aneps, anskok.

-ushort: anlicznik.

-bool anczysledzenie.

c)Kolejność wyczytywania danych:

-aneps, anpocz, ankon, anskok.

d)Warunki wprowadzanych danych:

-anpocz < ankon.

-aneps z zakresu (0;1).

-anskok > 0 i anskok < (anpocz – ankon) / 2.

e)Postać wypisywania wyników:

-Tabeli o kolumnach: x, S(x), Licznik zsumowanych wyrazów szeregu potęgowego.

f)Wzór:Obraz zawierający tekst

Opis wygenerowany automatycznie

g)Kod C#

static void anTablicowanieZmianyWartosciSzereguPotegowego(double anpocz, double ankon, double aneps, double anskok, ushort anlicznik, bool anczysledzenie)

{

Console.WriteLine("\t\t x\t S(x)\tLicznik zsumowanych wytazów szeregu potęgowego\n");

double ansuma;

while (anpocz <= ankon)

{

ansuma = anSumaSzereguPotegowego(anpocz, aneps, out anlicznik, anczysledzenie);

Console.WriteLine("\t\t{0, 5:F2}\t{1, 5:F2}\t{2,10}\n", anpocz, ansuma, anlicznik);

anpocz += anskok;

}

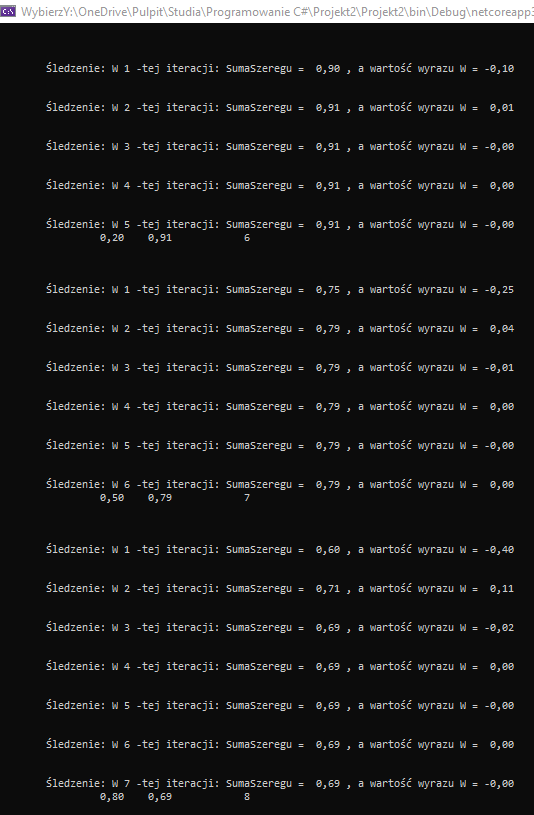
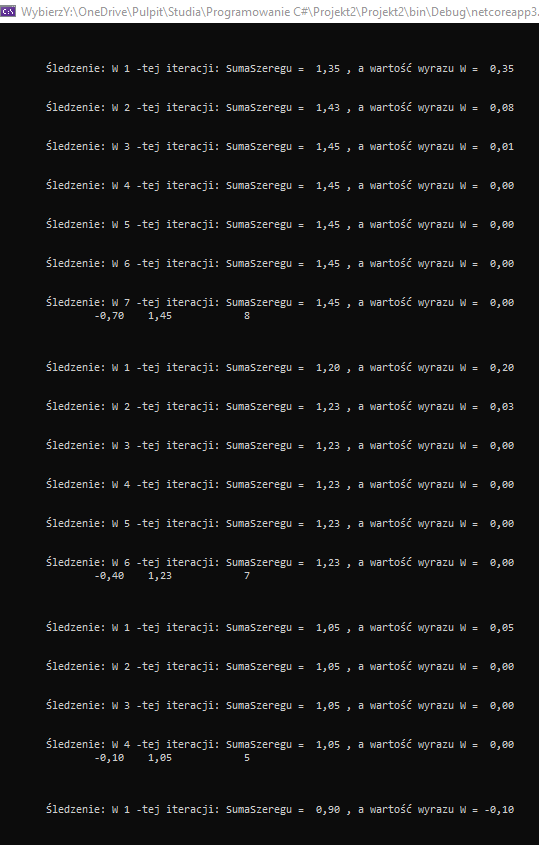
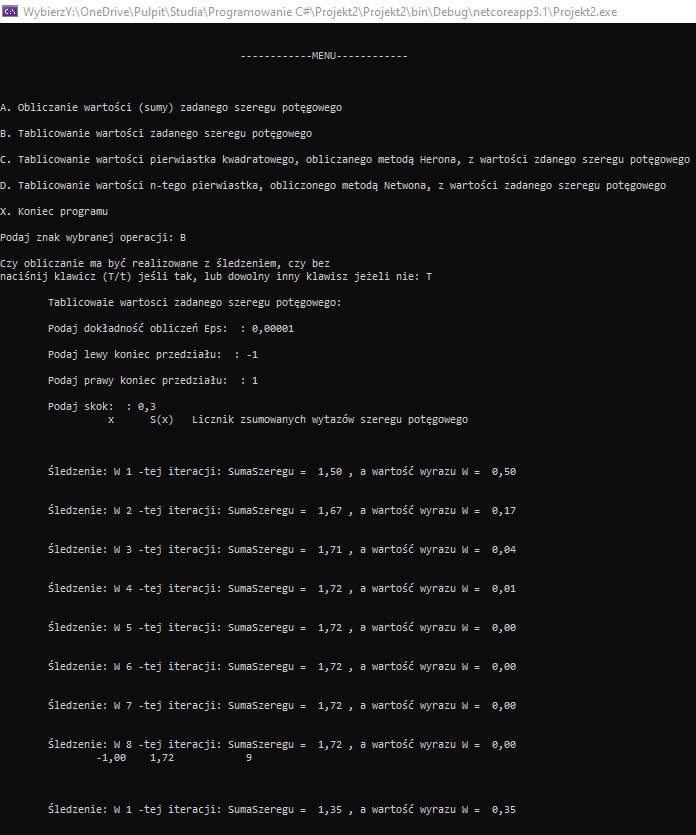
}

h)Dokumentacja testowania funkcjonalności:

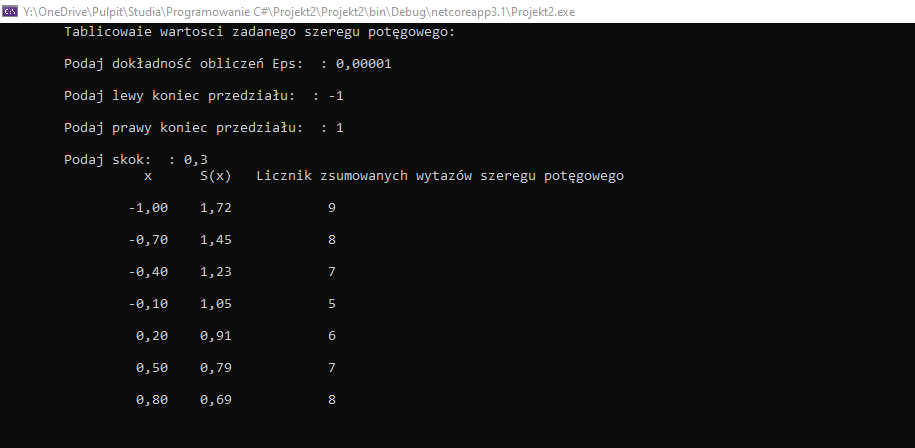
-Dane: aneps = 0.00001, anpocz = -1, ankon = 1, anskok = 0.3.

-Screen:

-Z śledzeniem:

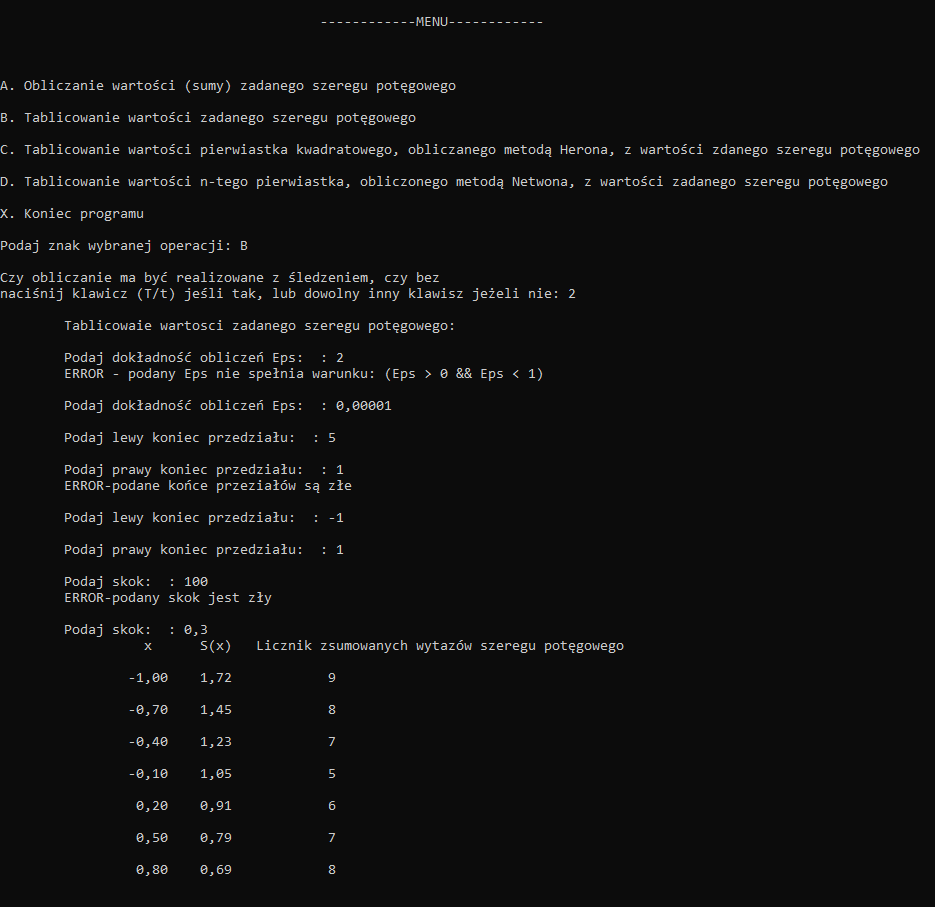


Bez śledzenia:

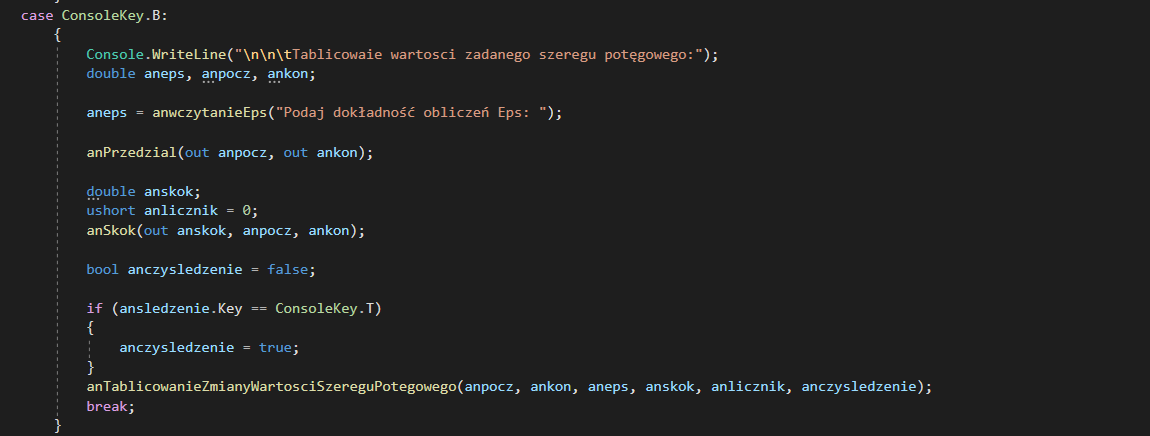


Błędne dane:

-Dane: aneps = 0.00001, anpocz = -1, ankon = 1, anskok = 0.3.



i)W main:



3. Tablicowanie wartości pierwiastka kwadratowego, obliczanego metodą Herona, z wartości zdanego szeregu potęgowego za pomocą metody: anTablicowaniepierwiastkaKwadratowego

a)Krótki opis funkcjonalności:

Tablicowanie wartości pierwiastka kwadratowego, obliczanego metodą Herona, z wartości zdanego szeregu potęgowego z użyciem dodatkowych metod: anSumaSzereguPotegowego, anSkok, anPrzedzial, anwczytanieEps, anPierwiatek2Stopnia (Zostaną one krótko opisane pod koniec dokumentacji).

b)Dane wejściowe wraz z typami:

-double: anpocz, ankon, aneps, anskok.

-ushort: anlicznik.

-bool anczysledzenie.

c)Kolejność wyczytywania danych:

-aneps, anpocz, ankon, anskok.

d)Warunki wprowadzanych danych:

-anpocz < ankon.

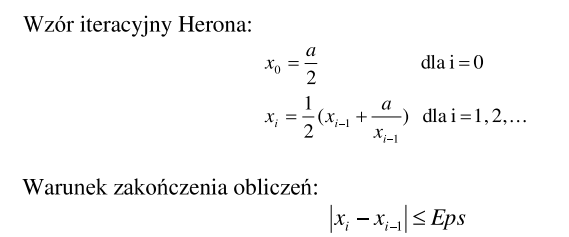
-aneps z zakresu (0;1).

-anskok > 0 i anskok < (anpocz – ankon) / 2.

e)Postać wypisywania wyników:

-Tabeli o kolumnach: x, S(x), Licznik zsumowanych wyrazów szeregu potęgowego, Pierwiastek drugiego stopnia.

f)Wzór:



g)Kod C#

static void anTablicowaniepierwiastkaKwadratowego(double anpocz, double ankon, double aneps, double anskok, ushort anlicznik, bool anczysledzenie)

{

Console.WriteLine("\t\t x\t S(x)\tLicznik zsumowanych wyrazów szeregu potęgowego\tPierwiastek drugiego stopnia\n");

double ansuma;

while (anpocz <= ankon)

{

ansuma = anSumaSzereguPotegowego(anpocz, aneps, out anlicznik, false);

Console.WriteLine("\t\t{0, 5:F2}\t{1, 5:F2}\t\t\t{2, 5:F2}\t\t\t\t{3, 5:F2}\n", anpocz, ansuma, anlicznik-1, anPierwiatek2Stopnia(ansuma, aneps, out anlicznik, anczysledzenie));

anpocz += anskok;

}

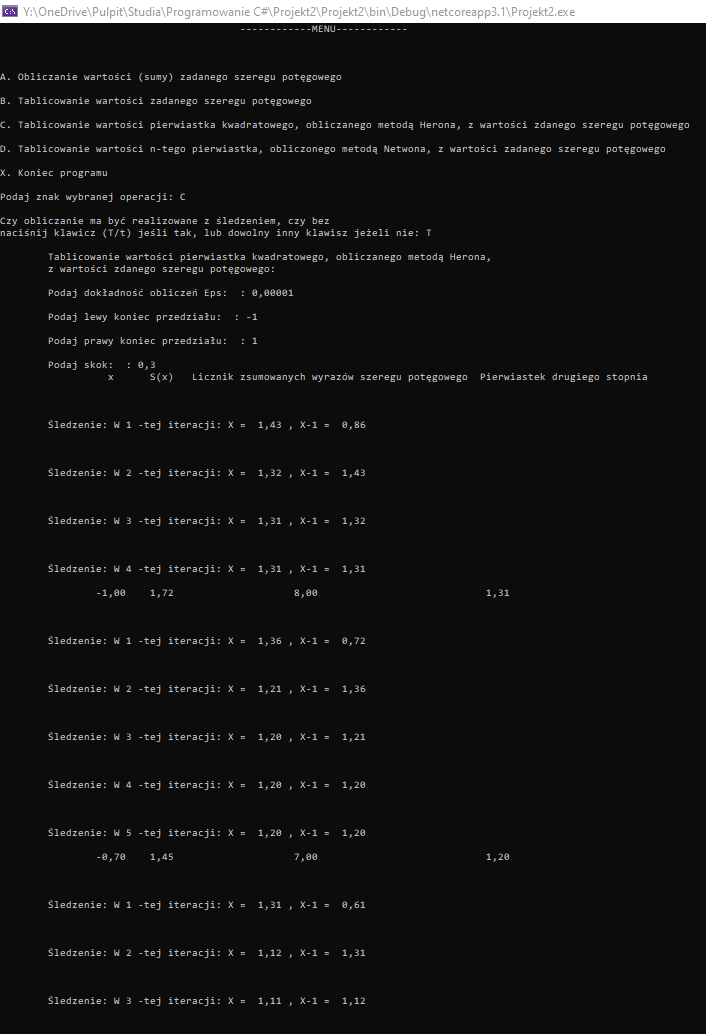
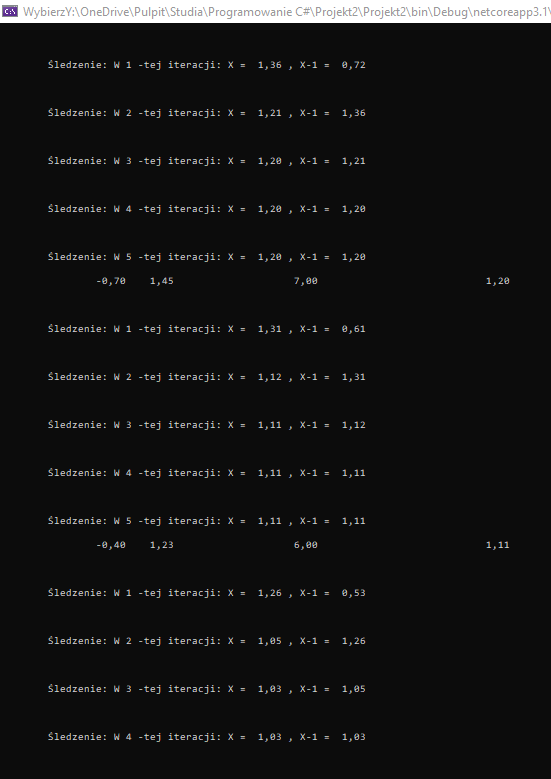
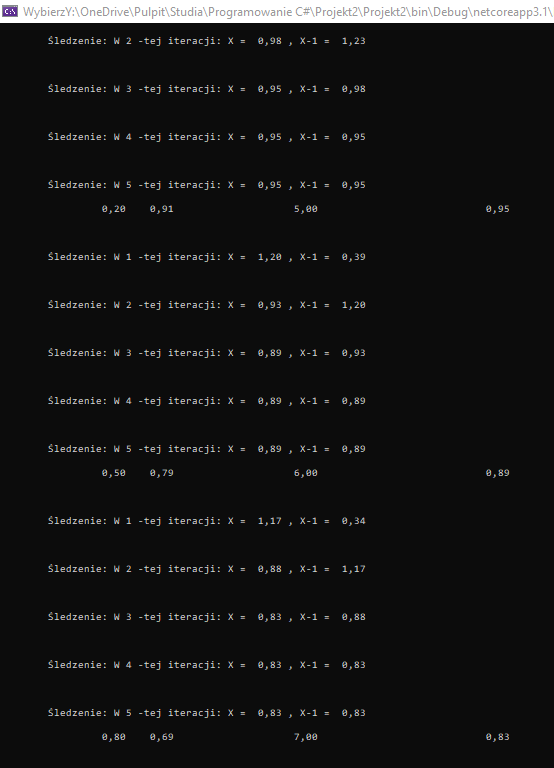
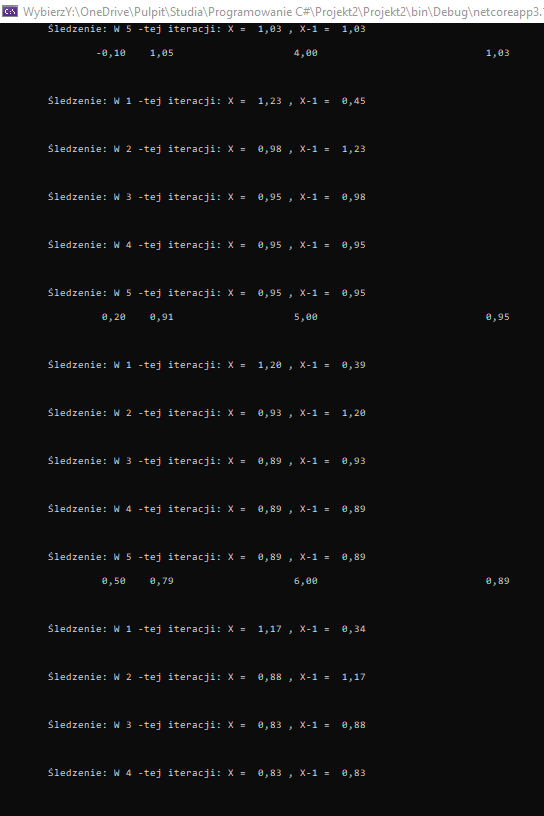
}

h)Dokumentacja testowania funkcjonalności:

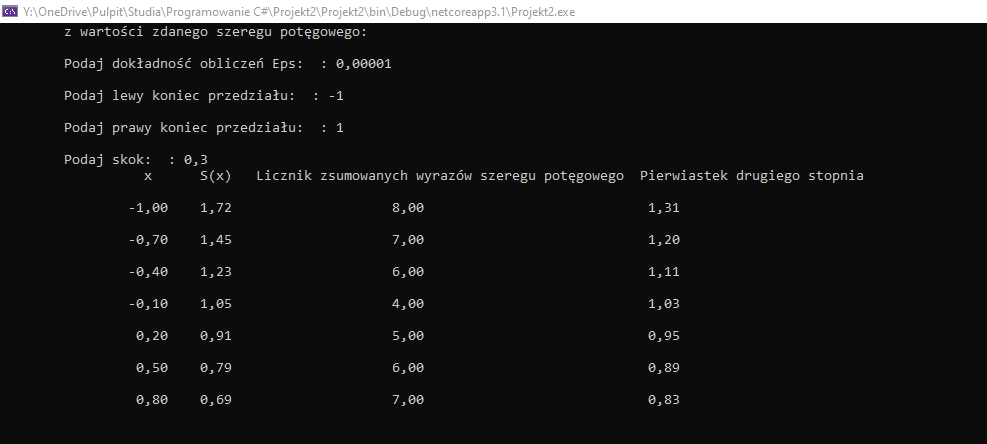
-Dane: aneps = 0.00001, anpocz = -1, ankon = 1, anskok = 0.3.

-Screen:

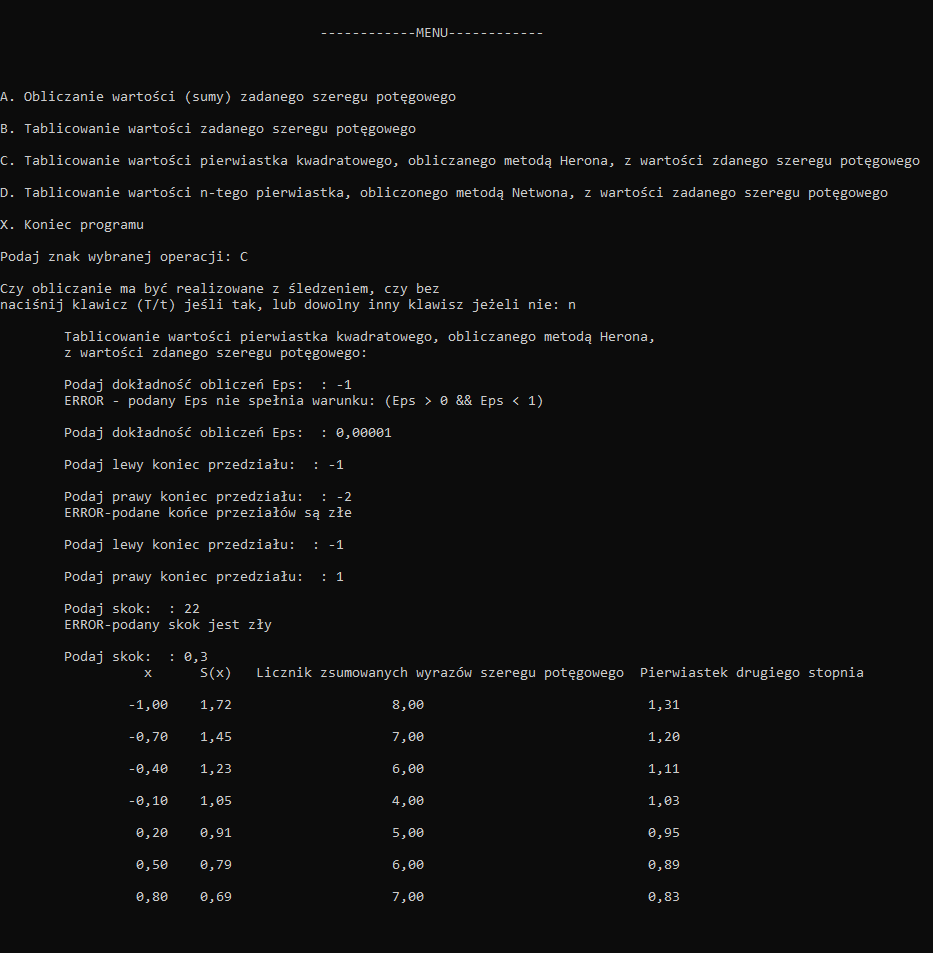
Z śledzeniem:



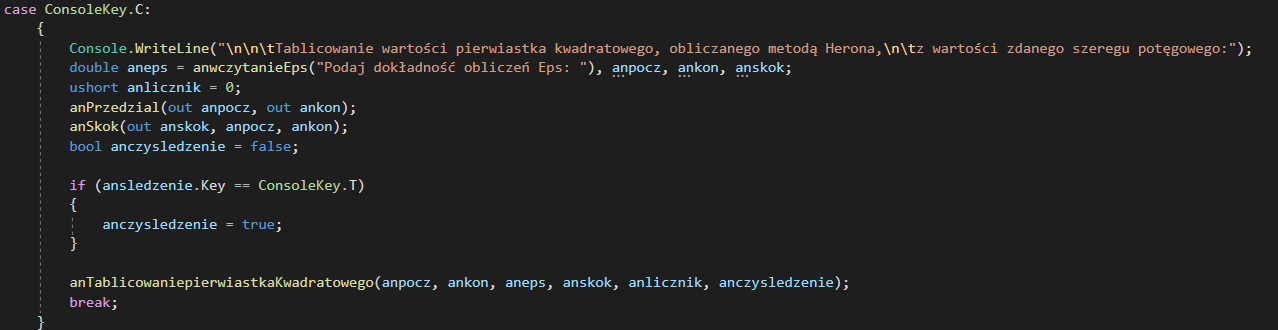
Bez śledzenia:



Błędne dane:



i)W main:



4. Tablicowanie wartości n-tego pierwiastka, obliczonego metodą Netwona, z wartości zadanego szeregu potęgowego:

a)Krótki opis funkcjonalności:

- Tablicowanie wartości n-tego pierwiastka, obliczanego metodą Netwona , z wartości zdanego szeregu potęgowego z użyciem dodatkowych metod: anSumaSzereguPotegowego, anSkok, anPrzedzial, anwczytanieEps, Tablicowanie\_nStopnia (Zostaną one krótko opisane pod koniec dokumentacji).

b)Dane wejściowe wraz z typami:

-double: anpocz, ankon, aneps, anskok.

-ushort: anlicznik, anstopien.

-bool anczysledzenie.

c)Kolejność wyczytywania danych:

-aneps, anstopien, anpocz, ankon, anskok.

d)Warunki wprowadzanych danych:

-anpocz < ankon.

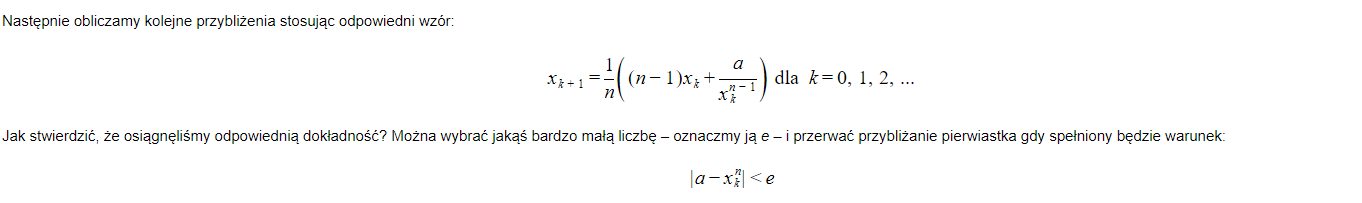
-aneps z zakresu (0;1).

-anskok > 0 i anskok < (anpocz – ankon) / 2.

-anstopien typu ushort.

e)Postać wypisywania wyników:

-Tabeli o kolumnach: x, S(x), Licznik zsumowanych wyrazów szeregu potęgowego, Pierwiastek n-tego stopnia.

f)Wzór:

g)Kod C#

static void Tablicowanie\_nStopnia(double anpocz, double ankon, double aneps, double anskok, ushort anlicznik, bool anczysledzenie, ushort anstopien)

{

Console.WriteLine("\t\t x\t S(x)\tLicznik zsumowanych wytazów szeregu\tPierwiastek {0} stopnia\n", anstopien);

double ansuma;

while (anpocz <= ankon)

{

ansuma = anSumaSzereguPotegowego(anpocz, aneps, out anlicznik, false);

Console.WriteLine("\t\t{0, 5:F2}\t{1, 5:F2}\t\t\t{2, 5:F2}\t\t\t\t{3, 5:F2}\n", anpocz, ansuma, anlicznik-1, anPierwiastekNStopnia(ansuma, anstopien, aneps, out anlicznik, anczysledzenie));

anpocz += anskok;

}

}

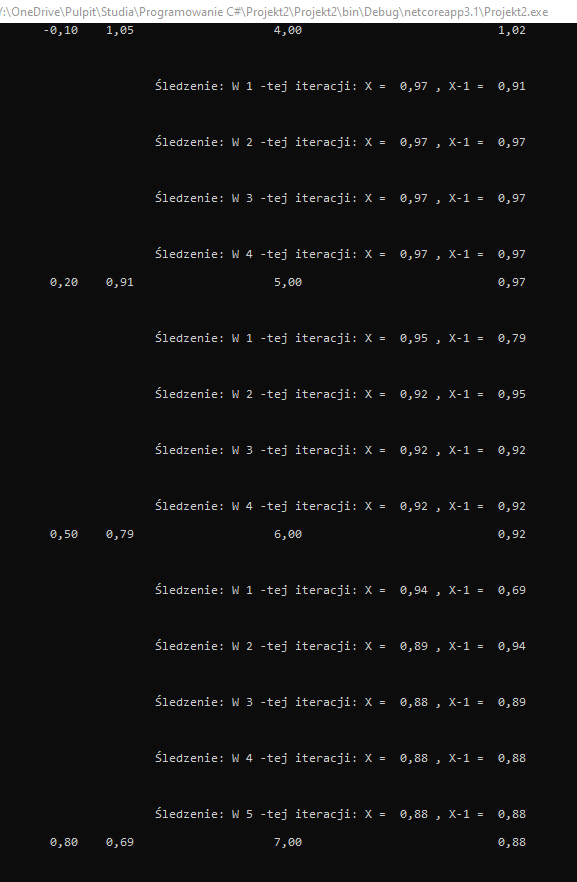
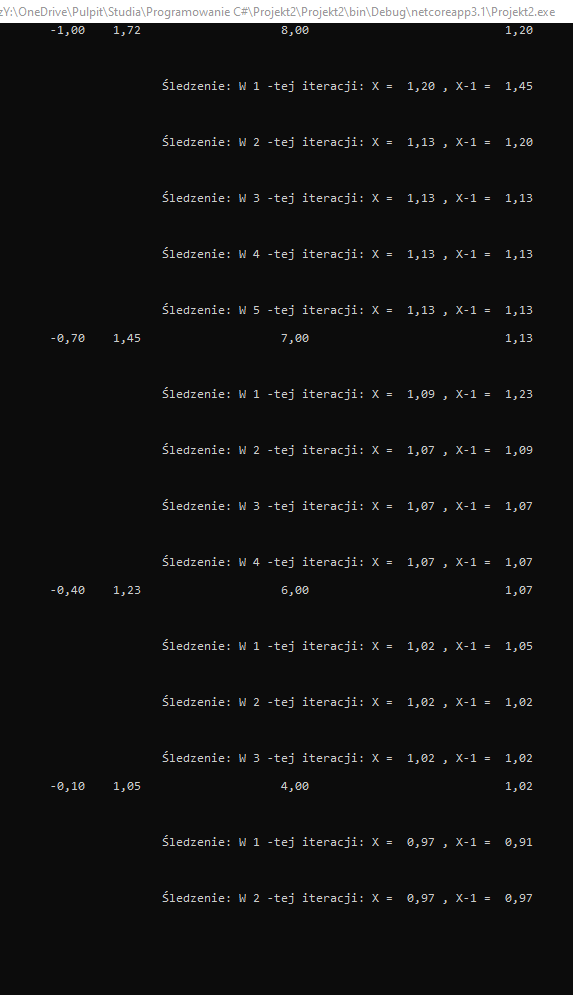
h)Dokumentacja testowania funkcjonalności:

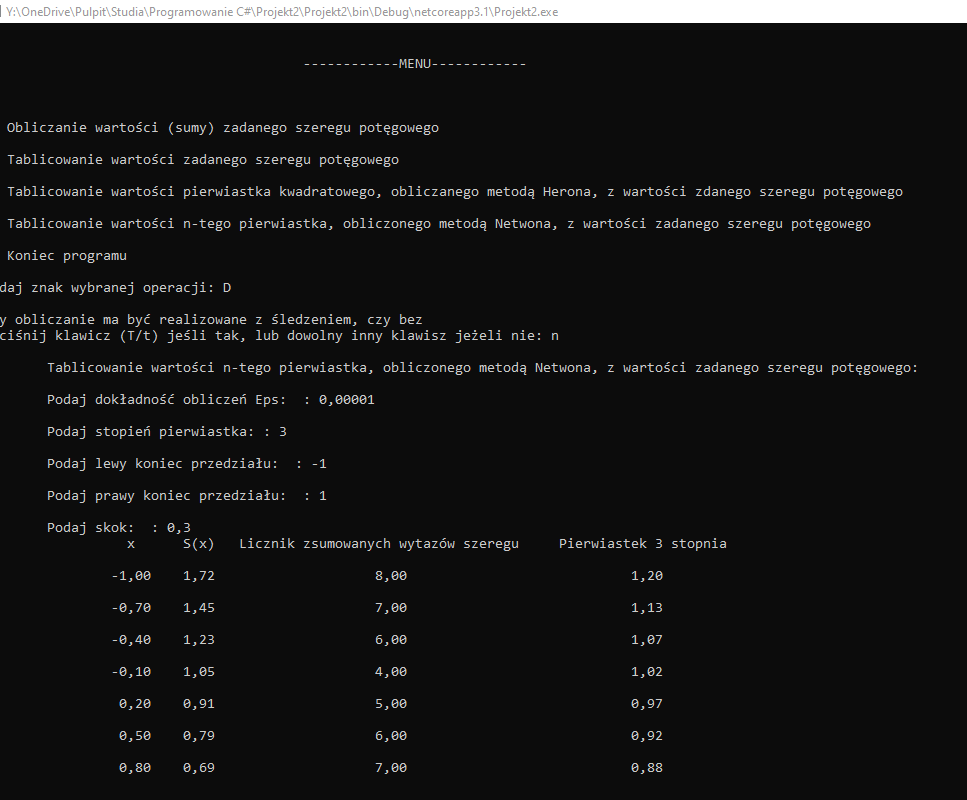
-Dane: aneps = 0.00001, anpocz = -1, ankon = 1, anskok = 0.3, anstopien = 3.

-Screen:

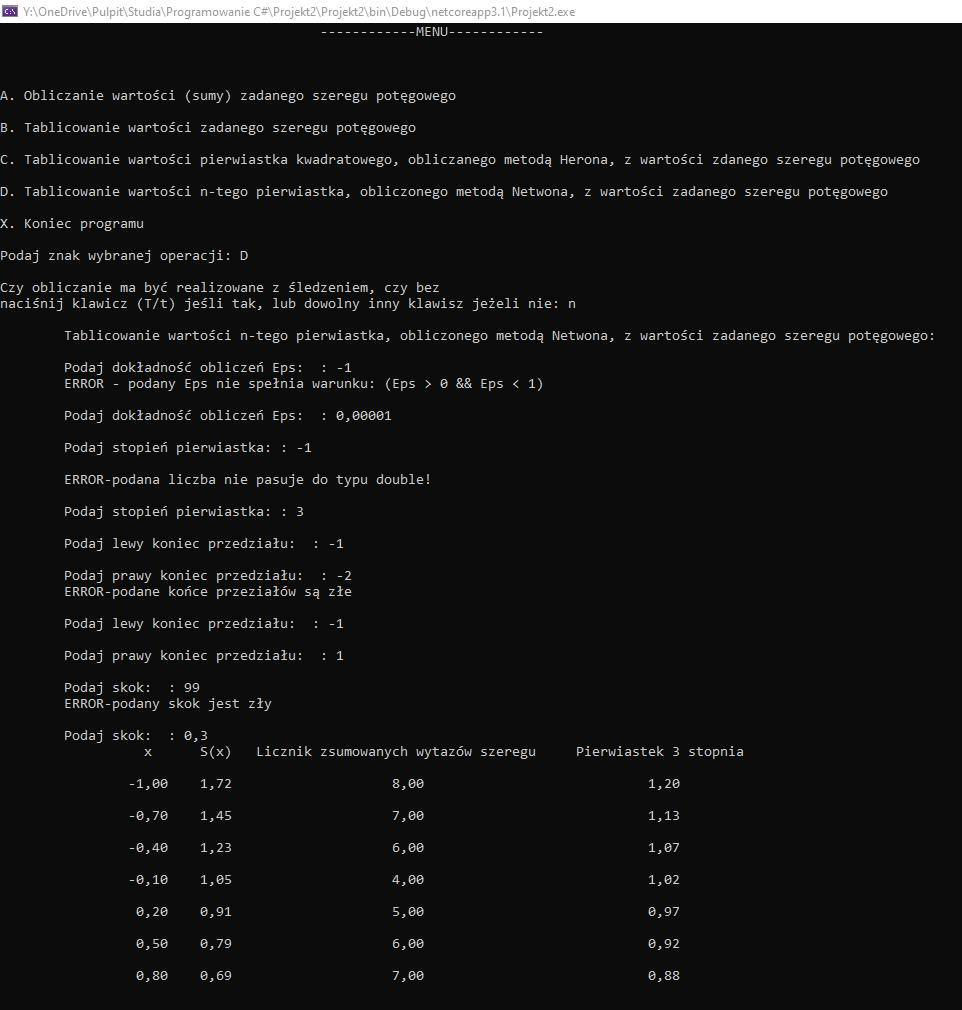
-Z śledzeniem: Obraz zawierający tekst

Opis wygenerowany automatycznie

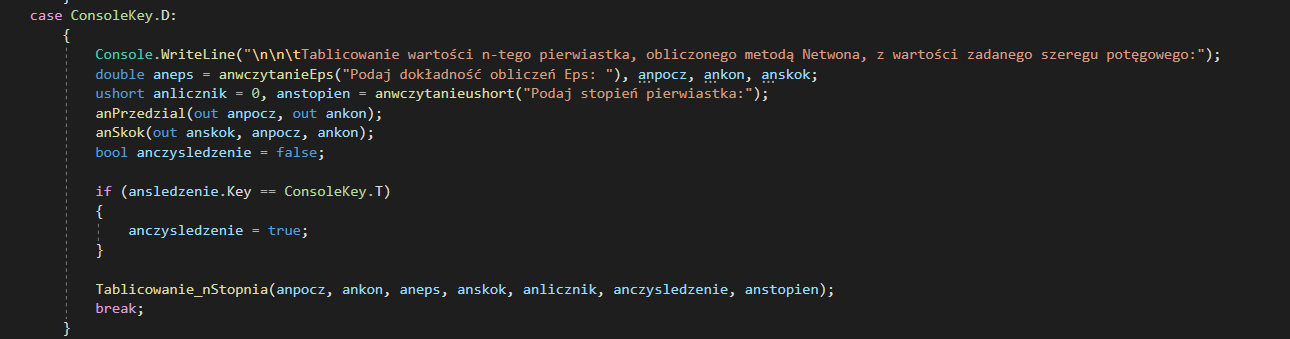


-Bez śledzenia:

-Błędne dane:



i)W main:



5.Metody pomocnicze:

a) anwczytanieEps

-Wczytuje liczbę w przedziale(0;1).

b)Kod:

static double anwczytanieEps(string antekst)

{

double aneps;

while (true)

{

Console.Write("\n\t{0} : ", antekst);

if (double.TryParse(Console.ReadLine(), out aneps) && aneps > 0 && aneps < 1)

{

break;

}

else

Console.WriteLine("\tERROR - podany Eps nie spełnia warunku: (Eps > 0 && Eps < 1) ");

}

return aneps;

}

b) anPrzedzial

-Wczytuje dwie liczby (końce przedziału). Warunek: anpocz < ankon;

b)Kod:

static void anPrzedzial(out double anpocz, out double ankon)

{

do

{

anpocz = anwczytaniedouble("Podaj lewy koniec przedziału: ");

ankon = anwczytaniedouble("Podaj prawy koniec przedziału: ");

if (anpocz >= ankon)

{

Console.WriteLine("\tERROR-podane końce przeziałów są złe");

}

} while (anpocz >= ankon);

}

c) anSkok

-Wczytuje liczbę (skok). Warunek: anskok > 0 i anskok < (anpocz – ankon) / 2)

Kod:

static void anSkok(out double anskok, double anpocz, double ankon)

{

do

{

anskok = anwczytaniedouble("Podaj skok: ");

if (anskok < 0 || anskok >= (ankon - anpocz) / 2)

{

Console.WriteLine("\tERROR-podany skok jest zły");

}

} while (anskok < 0 || anskok >= (ankon - anpocz) / 2);

}

d) anwczytaniedouble

-Wczytuje liczbę spełniającą warunki typu double.

Kod:

static double anwczytaniedouble(string antekst)

{

double anliczba;

while (true)

{

Console.Write("\n\t{0} : ",antekst);

if(double.TryParse(Console.ReadLine(), out anliczba))

{

break;

}else

Console.WriteLine("\n\tERROR-podana liczba nie pasuje do typu double!");

}

return anliczba;

}

e) anwczytanieushort

-Wczytuje liczbę spełniającą warunki typu ushort.

Kod:

static ushort anwczytanieushort(string antekst)

{

ushort anliczba;

while (true)

{

Console.Write("\n\t{0} : ", antekst);

if (ushort.TryParse(Console.ReadLine(), out anliczba))

{

break;

}

else

Console.WriteLine("\n\tERROR-podana liczba nie pasuje do typu double!");

}

return anliczba;

}

f) anpotega

-Liczy potęgę o określonej podstawie i wykłądniku.

Kod:

static double anpotega(double anpodstawa, ushort anwykladnik)

{

double anwynik = 1;

for (int i = 0; i < anwykladnik; i++)

{

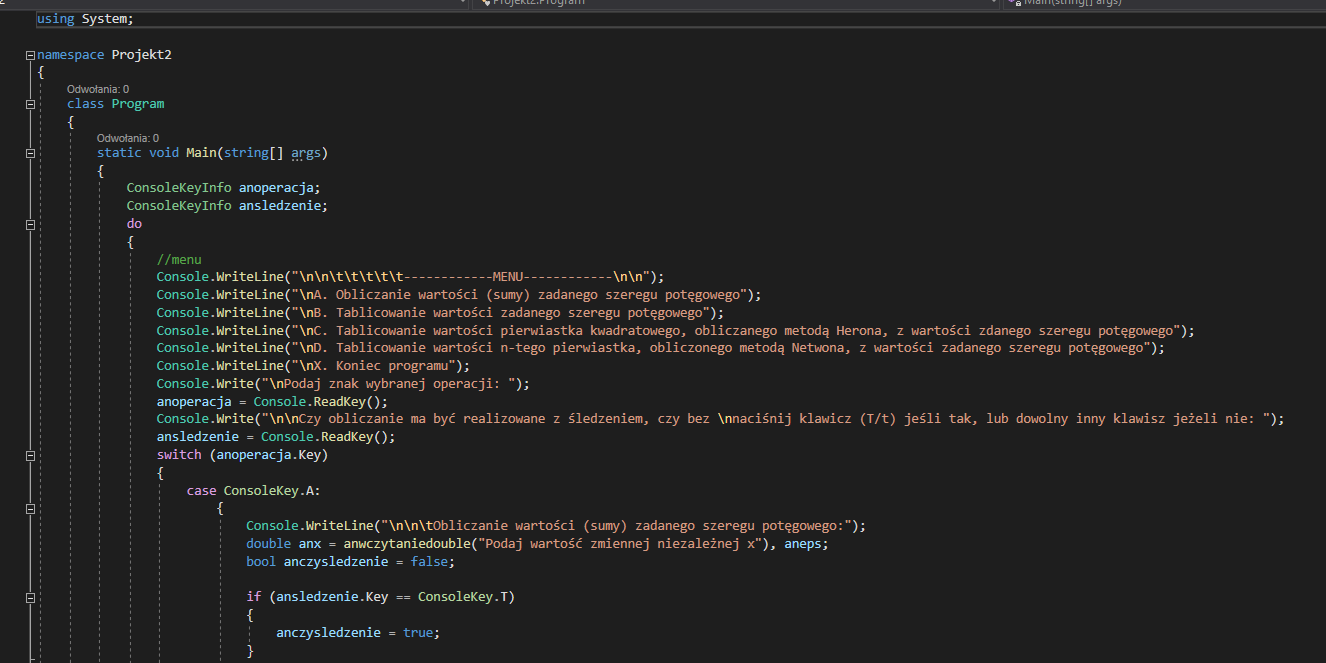
anwynik \*= anpodstawa;

}

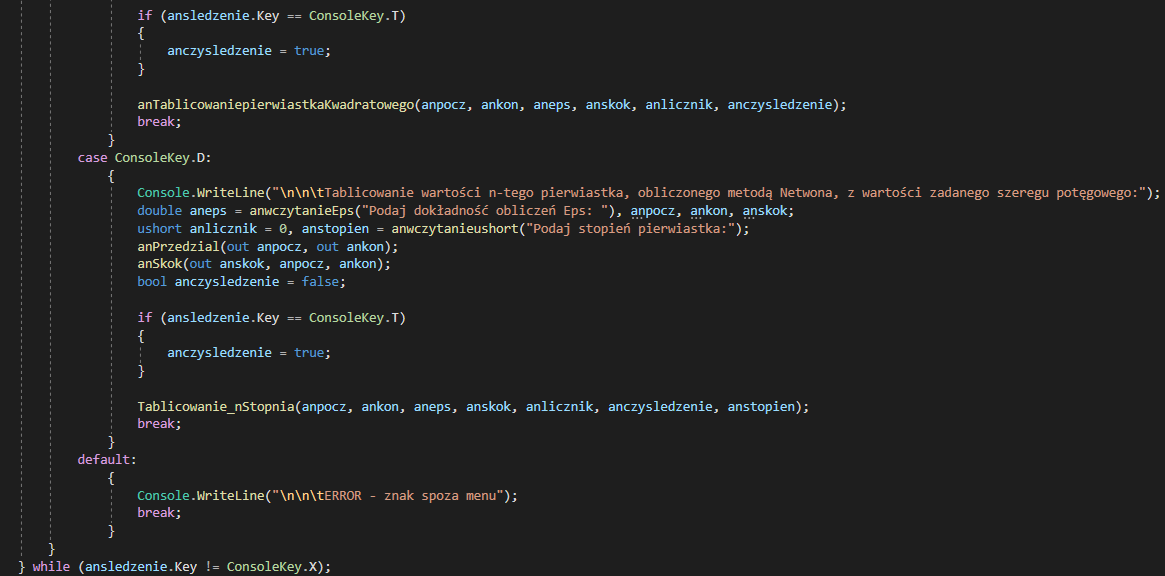
return anwynik;

}

6.Main:



Obraz zawierający tekst

Opis wygenerowany automatycznieWnioski końcowe:

-Swój projekt oceniam na 5.

-Natrafiłem na problem z metodą liczącą pierwiastek n-tego stopnia metodą Netwona.

Przy zapisie „(…)\*(1/n)” otrzymałem NaN, a przy zapisie: ”(…)/n” otrzymałem dobry wynik.

-Lepiej stworzyć main-a jak najkrótszym aby kod był bardziej przejrzysty.

-Brak sugestii.

7)SPRAWDZIAN Nr 2

a)Samoocena: Uważam, że dobrze napisałem sprawdzian.

b)Wyniki testów:

-Funkcjonalność: „Suma podzielników”.

case ConsoleKey.E:

{

Console.WriteLine("\n\n\tSuma podzielników:");

ushort ann = anwczytanieushort("Podaj liczbę wyrazów: ");

anSumaPodzielnikowWyswietlanie(ann);

break;

} }

Console.WriteLine("\n\t\tSuma podzielników podanych liczb to: {0}", ansuma);

}

static ushort ansumadzielnikow(ushort anliczba)

{

ushort ansuma = 0;

for (ushort i = 1; i < anliczba; i++)

{

if (anliczba % i == 0)

ansuma += i;

}

return ansuma;

}

static void anSumaPodzielnikowWyswietlanie(ushort ann)

{

ushort anliczba, ansuma = 0; ;

for (int i = 0; i < ann; i++)

{

Console.Write("\n\tPodaj liczbę nr {0}: ", i + 1);

if (!ushort.TryParse(Console.ReadLine(), out anliczba))

{

Console.WriteLine("\n\t\tERROR-podana liczba nie jest całkowita i/lub większa od zera");

i--;

}

else

{

ansuma += ansumadzielnikow(anliczba);

}

}

Console.WriteLine("\n\t\tSuma podzielników podanych liczb to: {0}", ansuma);

}

static ushort ansumadzielnikow(ushort anliczba)

{

ushort ansuma = 0;

for (ushort i = 1; i < anliczba; i++)

{

if (anliczba % i == 0)

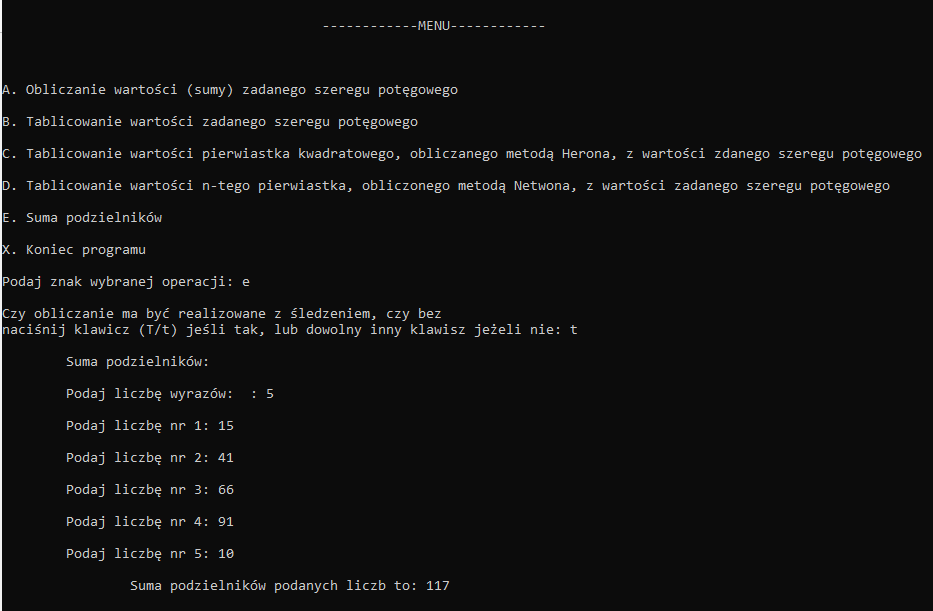
ansuma += i;

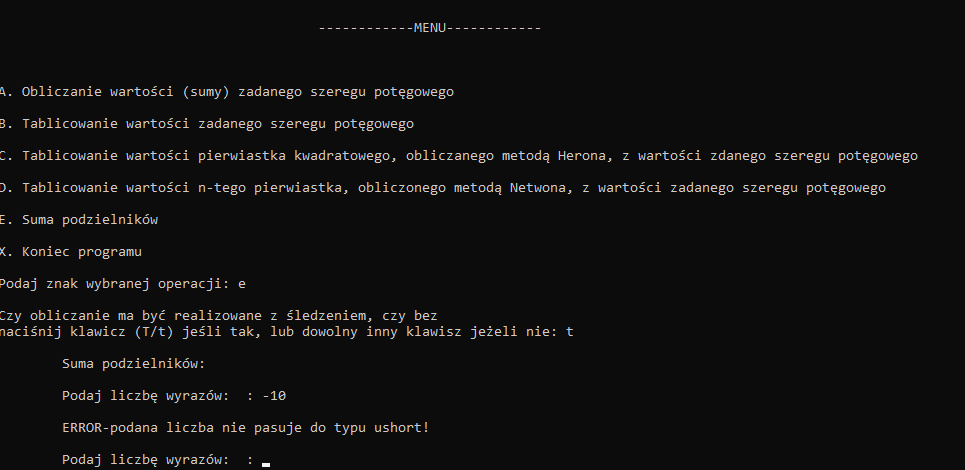
}

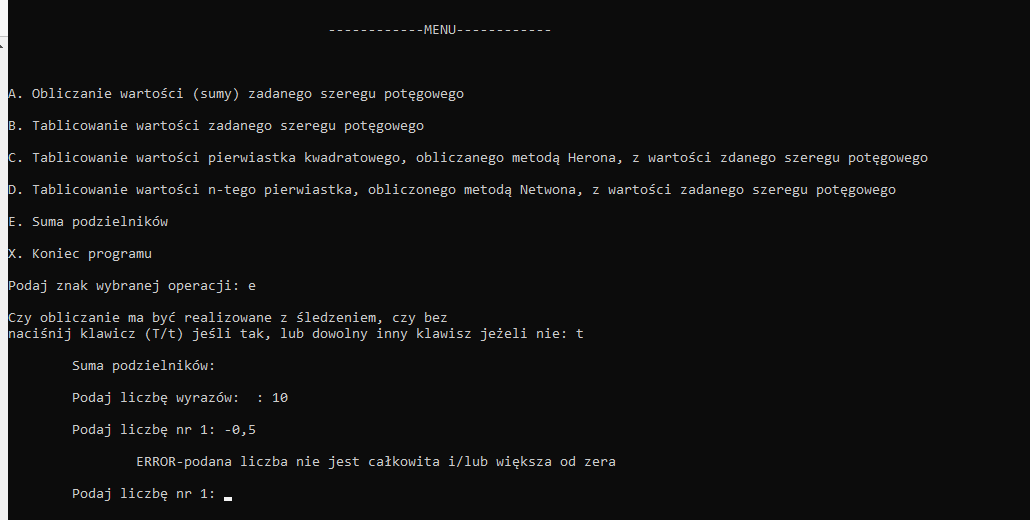
return ansuma;

}

-I zestaw danych testowych : 2, 12, 12.

-II zestaw danych testowych: 5, 15, 41, 66, 91, 10.

-I zestaw nieprawidłowych danych testowych: -10.

-II zestaw nieprawidłowych danych testowych: 10, -0.5 .

c) Wyniki testów:

-Funkcjonalność: Palindrom.

case ConsoleKey.F:

{

Console.WriteLine("\n\n\tPalindrom:");

Console.Write("\n\tPodaj ciąg znaków: ");

string anslowo = Console.ReadLine();

if(anczypalindrom(anslowo))

Console.WriteLine("\n\t\tPodany ciąg znaków jest palindromem");

else

Console.WriteLine("\n\t\tPodany ciąg znaków nie jest palindromem");

break;

}

static bool anczypalindrom(string anslowo)

{

string anlustro = "";

for (int i = 0; i < anslowo.Length; i++)

{

anlustro += anslowo[anslowo.Length - i - 1];

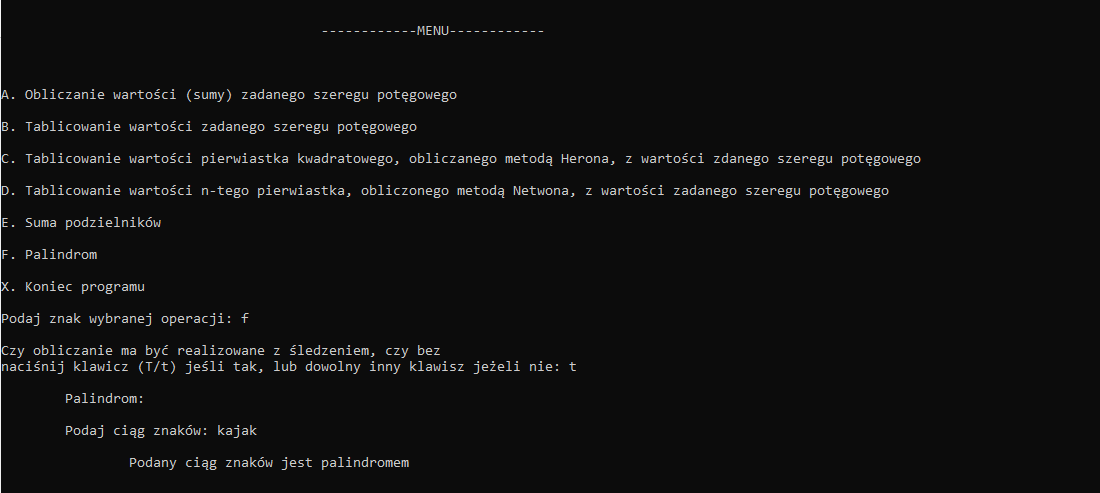
}

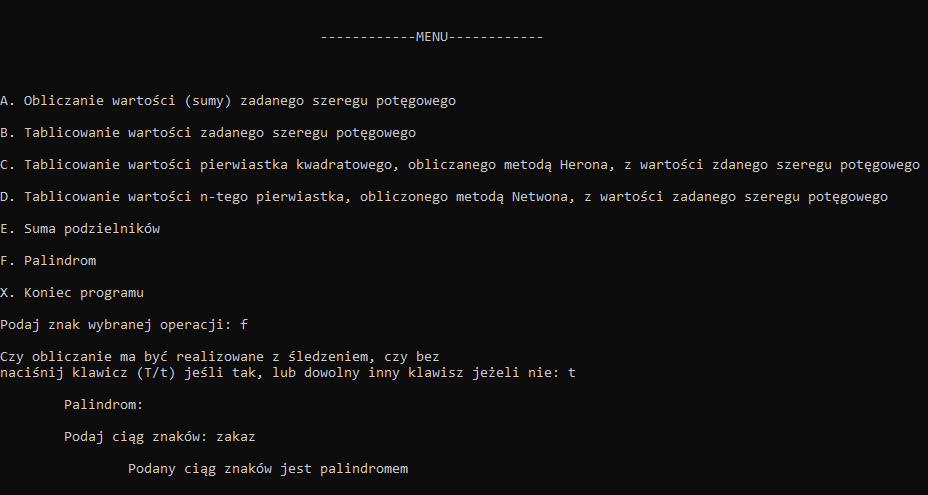
if (anslowo == anlustro)

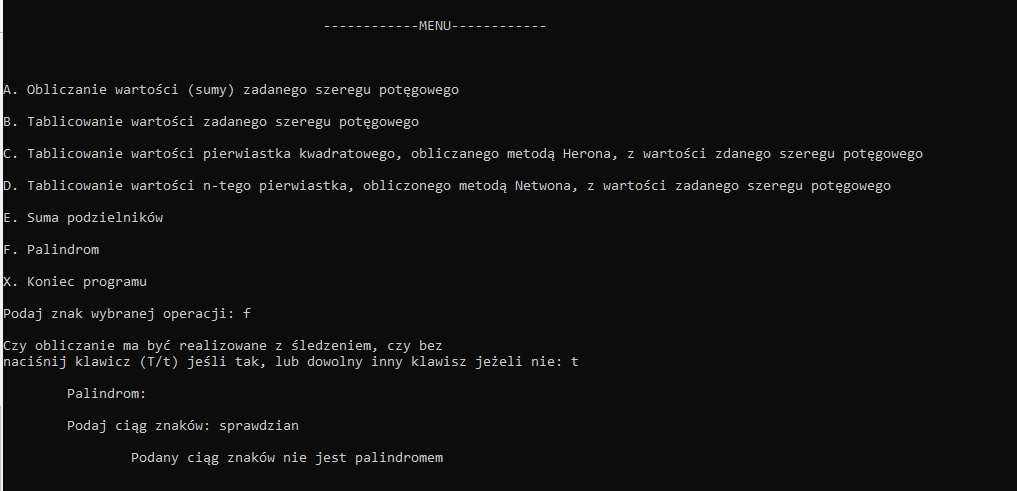
return true;

return false;

}

-I zestaw danych testowych: kajak.

-II zestaw danych testowych: zakaz.

-III zestaw danych testowych: sprawdzian. 

-IV zestaw danych testowych: koniec. Obraz zawierający tekst

Opis wygenerowany automatycznie