Prowadzący zajęcia  
dr inż. Andrzej Urbański

**Laboratoria nr1**

**Programowanie Niskopoziomowe**

**Proste programy języka C**

Wojciech Regulski  
Informatyka(WI) I1  
nr 132312

Zadanie nr 1.

Opracować program wczytujący 3 liczby typu double i wyprowadzający największą z tych liczb.

#include <stdio.h>

int main()

{

double a,b,c;

printf("Podaj a: ");

scanf("%lf",&a);

printf("Podaj b: ");

scanf("%lf",&b);

printf("Podaj c: ");

scanf("%lf",&c);

if (a>=b)

{

if(a>=c)

printf("Najwieksza liczba jest a=%lf", a);

else

printf("Najwieksza liczba jest c=%lf", c);

}

else

{

if(b>=c)

printf("Najwieksza liczba jest b=%lf", b);

else

printf("Najwieksza liczba jest c=%lf", c);

}

return 0;

}

Przy zaledwie trzech liczbach, sprawdzenie, która jest największa, ogranicza się do zaledwie kilku instrukcji warunkowych. Należy pamiętać, aby nie przeoczyć żadnej z możliwych opcji.

Zadanie nr 2.

Wyznaczenie wartości wyrażenia a\*(a+b)/(a+b)2, gdzie a i b to liczby całkowite różne od 0.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int a,b;

do

{

system("cls"); //czyszczenie konsoli w razie kolejnego wczytywania danych

printf("Podaj a: ");

scanf("%d",&a);

printf("Podaj b: ");

scanf("%d",&b);

}

while(a==0 || b==0 || a==-b); //zabezpieczenie przed dzieleniem przez 0

printf("Wynik: %f",(float)a/(a+b));

return 0;

}

W tym zadaniu wprowadziłem drobną optymalizację upraszczając „a+b” z licznika i mianownika. Nie dopuszczam również w programie dzielenia przez zero, każąc wpisywać liczby a i b jeszcze raz w przypadku, gdy przy podanych danych wejściowych miałoby to miejsce.

Zadanie nr 3.

Opracować program, który wylicza wartość wyrażenia:

Dane: a, b typu float

#include <stdio.h>

int main()

{

float a,b;

printf("Podaj a: ");

scanf("%f",&a);

printf("Podaj b: ");

scanf("%f",&b);

printf("Wynik: %f",4\*a+2\*(a-b-1)/(a\*a+b\*b+1));

return 0;

}

Ewentualna trudność tego zadania mogła wynikać z konieczności zapisania w kodzie działań na ułamkach.

Zadanie nr 4.

Opracować program, który oblicza wartość wyrażenia:

Dane: m, n typu int

#include <stdio.h>

int main()

{

int m,n;

do

{

printf("Podaj m: ");

scanf("%d",&m);

printf("Podaj n: ");

scanf("%d",&n);

}

while((m+n)==5);

printf("Wynik: %f",(float)(5+m-n)/(m+n-5));

return 0;

}

W tym zadaniu zastosowałem optymalizację polegającą na znacznym uproszczeniu wyrażenia matematycznego do wyliczenia. Zabezpieczyłem również program przed działaniem na liczbach, które doprowadzą do dzielenia przez zero. Do poprawnego działania programu potrzebne było też rzutowanie z typu int na float, aby wynik nie miał odjętej części ułamkowej.Zadanie nr 5.

Opracować program, który wylicza wartość wyrażenia:

Dane : x typu double

#include <stdio.h>

#include <math.h>

int main()

{

double x;

printf("Podaj x: ");

scanf("%lf",&x);

printf("Wynik: %lf",(sqrt((pow(sin(x\*x\*x),2)+1.25)/(pow(cos(x\*x),3)+1.25)))/(log(pow(tan(x+2),2))+2.5));

return 0;

}

To zadanie pokazuje jak skomplikowanie w kodzie programu może wyglądać wyrażenie matematyczne. Przy takich wyrażeniach bardzo łatwo postawić nawias w złym miejscu i całe wyrażenie traci sens. Do tego zadania potrzebna była biblioteka math.h, dzięki której można było wykonać obliczenie wartości funkcji trygonometrycznych i logarytmu.

Zadanie nr 6.

Opracować program zamiany temperatury wyrażonej w stopniach Celsjusza na stopnie Fahrenheita i odwrotnie.

Program powinien pytać o kierunek konwersji, wczytywać temperaturę i wyprowadzać wartość przeliczoną wraz z jednostką.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int wybor;

float T;

do

{

system("cls");

printf("1 - Zamiana ze stopni Celcjusza na Fahrenheita\n");

printf("2 - Zamiana ze stopni Fahrenheita na Celcjusza\n");

printf("Wybierz opcje 1 lub 2: ");

scanf("%d",&wybor);

}

while(wybor!=1 && wybor!=2);

printf("\nPodaj wartosc temperatury: ");

scanf("%f",&T);

switch(wybor)

{

case 1:

{

printf("\nWynik: %.2f stopni Fahrenheita", 1.8\*T+32);

break;

}

case 2:

{

printf("\nWynik: %.2f stopni Celsjusza", 0.55555556\*(T-32));

break;

}

}

return 0;

}

Wybór opcji zrealizowany został w pętli do while, która nie pozwala wybrac innej opcji niż dopuszczalne. Na podstawie wpisanej przez użytkownika liczby, konstrukcja swich dopuszcza do wykonania odpowiednią procedurę obliczania temperatury w drugiej jednostce.

Zadanie nr 7.

Sprawdzić rodzaj figury geometrycznej (prostokąt, kwadrat) na podstawie współrzędnych wszystkich wierzchołków podanych na wejście programu.

#include <stdio.h>

#include <math.h>

int main()

{

float x1,x2,x3,x4,y1,y2,y3,y4,xk,yk,dl1,dl2,dl3,dl4,dlk1,dlk2,dlk3,dlk4;

printf("Jako kolejny punkt przyjmuj kolejny wedlug wskazowej zegara lub odwrotnie do ruchu wskazowek zegara, np.\n");

printf("1-----2\n| |\n4-----3\n\n");

printf("Podaj x1: "); scanf("%f",&x1);

printf("Podaj y1: "); scanf("%f",&y1);

printf("Podaj x2: "); scanf("%f",&x2);

printf("Podaj y2: "); scanf("%f",&y2);

printf("Podaj x3: "); scanf("%f",&x3);

printf("Podaj y3: "); scanf("%f",&y3);

printf("Podaj x4: "); scanf("%f",&x4);

printf("Podaj y4: "); scanf("%f",&y4);

xk=(x1+x2+x3+x4)/4;

yk=(y1+y2+y3+y4)/4;

dlk1=pow(xk-x1,2)+pow(yk-y1,2);

dlk2=pow(xk-x2,2)+pow(yk-y2,2);

dlk3=pow(xk-x3,2)+pow(yk-y3,2);

dlk4=pow(xk-x4,2)+pow(yk-y4,2);

dl1=pow(x2-x1,2)+pow(y2-y1,2);

dl2=pow(x2-x3,2)+pow(y2-y3,2);

dl3=pow(x3-x4,2)+pow(y3-y4,2);

dl4=pow(x4-x1,2)+pow(y4-y1,2);

if((fabs(dlk1-dlk2)<0.0001) && (fabs(dlk1-dlk3)<0.0001) && (fabs(dlk1-dlk4)<0.0001)) {

if((fabs(dl1-dl2)<0.0001) && (fabs(dl1-dl3)<0.0001) && (fabs(dl1-dl4)<0.0001))

printf("Figura jest kwadratem");

else

printf("Figura jest prostokatem");

} else

printf("Figura inna niz kwadrat i prostokat");

return 0;

}

Do określenia figury została wykorzystana właściwość prostokątów co do ich wpisania w okrąg – odległości między środkiem okręgu a wierzchołkami prostokąta (dlk1-4) są takie same. Gdy dodatkowo długości boków są równe, to prostokąt jest kwadratem. Odległości nie są liczone dokładnie (nie są pierwiastkowane), gdyż nie jest to konieczne. Konstrukcje typu „fabs(dlk1-dlk2)<0.0001” są zastosowane z powodu, że licząc w systemie binarnym mogą się pojawić delikatne różnice między teoretycznie równymi liczbami.

Zadanie nr 8.

Obliczyć punkt przecięcia dwóch prostych oraz jego odległość od początku układy współrzędnych.

#include <stdio.h>

#include <math.h>

int main()

{

float a1,a2,b1,b2,x,y,odl;

printf("a1=");

scanf("%f",&a1);

printf("b1=");

scanf("%f",&b1);

printf("a2=");

scanf("%f",&a2);

printf("b2=");

scanf("%f",&b2);

if(a1==a2)

{

printf("Proste sa rownolegle - nie przecinaja sie");

return 0;

}

x=(b2-b1)/(a1-a2);

y=a1\*x+b1;

odl=sqrt(x\*x+y\*y);

printf("\nProste przecinaja sie w punkcie(%.4f,%.4f), ktory lezy %.4f jednostki dlugosci od poczatku ukladu wspolrzednych",x,y,odl);

return 0;

}

Przełożenie podstawowej wiedzy z matematyki na język programowania.

Zadanie nr 9.

Wyznaczenie pierwiastków równania kwadratowego ax2+bx+c=0, gdzie a, b, c to dowolne liczby całkowite.

#include <stdio.h>

#include <math.h>

int main()

{

int a,b,c;

float delta;

printf("a= ");

scanf("%d",&a);

printf("b= ");

scanf("%d",&b);

printf("c= ");

scanf("%d",&c);

delta = b\*b-4\*a\*c;

if(delta>0)

{

printf("Pierwiastki rownania: x1=%f, x2=%f",(-b-sqrt(delta))/(2\*a),(-b+sqrt(delta))/(2\*a));

}

if(delta==0)

{

printf("Pierwiastek rownania: x=%f",(float)-b/(2\*a));

}

if(delta<0)

{

float x=(float)-b/(2\*a);

float i=sqrt(-delta)/(2\*a);

printf("Pierwiastki rownania: x1=%f+%fi, x2=%f-%fi",x,i,x,i);

}

return 0;

}

Przełożenie podstawowej wiedzy z algebry na język programowania.

Zadanie nr 10.

Obliczanie obwodów i pól figur płaskich (koło, kwadrat, prostokąt, trójkąt, trapez), gdzie rodzaj figury i wymagane dane wykorzystywane podczas obliczeń definiuje użytkownik.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int wybor;

do

{

system("cls");

printf("1-kolo\n2-kwadrat\n3-prostokat\n4-trojkat\n5-trapez\nWybor: ");

scanf("%d",&wybor);

}while(wybor!=1 && wybor!=2 && wybor!=3 && wybor!=4 && wybor!=5);

switch(wybor)

{

case 1:

{

float r;

printf("\nr=");

scanf("%f",&r);

printf("obwod=%f\n",3.14159265\*2\*r);

printf("pole=%f",3.14159265\*r\*r);

break;

}

case 2:

{

float a;

printf("\na=");

scanf("%f",&a);

printf("obwod=%f\n",4\*a);

printf("pole=%f",a\*a);

break;

}

case 3:

{

float a,b;

printf("\na=");

scanf("%f",&a);

printf("b=");

scanf("%f",&b);

printf("obwod=%f\n",2\*a+2\*b);

printf("pole=%f",a\*b);

break;

}

case 4:

{

float a,b,c,p;

do

{

printf("\na=");

scanf("%f",&a);

printf("b=");

scanf("%f",&b);

printf("c=");

scanf("%f",&c);

}

while(!(((a+b)>c) && ((a+c)>b) && ((b+c)>a)));

p=(a+b+c)/2;

printf("obwod=%f\n",a+b+c);

printf("pole=%f",sqrt(p\*(p-a)\*(p-b)\*(p-c)));

break;

}

case 5:

{

float a,b,c,d,h;

printf("\na(podstawa)=");

scanf("%f",&a);

printf("b(podstawa)=");

scanf("%f",&b);

printf("c(ramie)=");

scanf("%f",&c);

printf("d(ramie)=");

scanf("%f",&d);

printf("h(wysokosc)=");

scanf("%f",&h);

printf("obwod=%f\n",a+b+c+d);

printf("pole=%f",0.5\*((a+b)\*h));

break;

}

}

return 0;

}

Pętla do while nie pozwala wybrać innej figury niż przewidziana. Wykonanie odpowiednich instrukcji do figury gwarantuje konstrukcja switch. Przy trójkącie przed działaniami sprawdzana jest możliwość istnienia trójkąta dla podanych boków.

Zadanie nr 11.

Kalkulator udostępniający następujące operacje: potęgowanie, wyznaczanie odwrotności danej liczby, logarytm naturalny danej liczby, sinus, tangens z danej liczby.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int wybor;

do

{

system("cls");

printf("1-potegowanie\n2-odwrotnosc\n3-logarytm naturalny\n4-sinus\n5-tangens\nWybor: ");

scanf("%d",&wybor);

}while(wybor!=1 && wybor!=2 && wybor!=3 && wybor!=4 && wybor!=5);

switch(wybor)

{

case 1:

{

float x,y;

printf("\npodstawa=",&x);

scanf("%f",&x);

printf("wykladnik=");

scanf("%f",&y);

printf("wynik=%f",pow(x,y));

break;

}

case 2:

{

float x;

printf("\nx=",&x);

scanf("%f",&x);

printf("wynik=%f",1/x);

break;

}

case 3:

{

float x;

printf("\nx=",&x);

scanf("%f",&x);

printf("wynik=%f",log(x));

break;

}

case 4:

{

float x;

printf("\nx=",&x);

scanf("%f",&x);

printf("wynik=%f",sin(x));

break;

}

case 5:

{

float x;

printf("\nx=",&x);

scanf("%f",&x);

printf("wynik=%f",tan(x));

break;

}

}

return 0;

}

Wykorzystanie biblioteki math.h

Zadanie nr 12.

Napisz program czytania trzech liczb i sprawdzania, czy mogą być one długościami boków jakiegoś trójkąta.

#include <stdio.h>

#include <stdlib.h>

int main()

{

float a,b,c;

printf("a=");

scanf("%f",&a);

printf("b=");

scanf("%f",&b);

printf("c=");

scanf("%f",&c);

if(((a+b)>c) && ((a+c)>b) && ((b+c)>a))

printf("boki moga utworzyc trojkat");

else

printf("boki nie moga utworzyc trojkata");

return 0;

}

Przełożenie podstawowej wiedzy z matematyki na język programowania.

Zadanie nr 13.

Wyznaczenie średniej arytmetycznej, geometrycznej n naturalnych liczb wejściowych.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

int n,i;

float suma=0,iloczyn=1;

printf("n=");

scanf("%d",&n);

float tab[n];

for(i=0;i<n;i++)

{

printf("liczba %d.=",i+1);

scanf("%f",&tab[i]);

suma+=tab[i];

iloczyn\*=tab[i];

}

printf("\nSrednia arytmetyczna=%f",suma/n);

printf("\nSrednia geometryczna=%f",pow(iloczyn,(float)1/n));

return 0;

}

Przełożenie podstawowej wiedzy z matematyki na język programowania.

Zadanie nr 14.

Napisać program losujący n liczb z przedziału od 0 do m.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,m,i;

srand(time(0));

do

{

system("cls");

printf("n=");

scanf("%d",&n);

printf("m=");

scanf("%d",&m);

}while(m<0);

m++;

for(i=0;i<n;i++)

printf("%d. liczba: %d\n",i+1,rand()%m);

return 0;

}

Przedział losowania od 0 do m uzyskujemy przez odczyt reszty z dzielenia liczby pseudolosowej przez m+1.

Zadanie nr 15.

Zbadaj współliniowość trzech punktów na podstawie podanych ich współrzędnych.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

float x1,x2,x3,y1,y2,y3;

printf("Podaj x1: ");

scanf("%f",&x1);

printf("Podaj y1: ");

scanf("%f",&y1);

printf("Podaj x2: ");

scanf("%f",&x2);

printf("Podaj y2: ");

scanf("%f",&y2);

printf("Podaj x3: ");

scanf("%f",&x3);

printf("Podaj y3: ");

scanf("%f",&y3);

if(x1==x2 && x1==x3)

{

printf("\npunkty sa wspolliniowe");

return 0;

}

if(x1==x2 || x1==x3 || x2==x3)

{

printf("\npunkty nie sa wspolliniowe");

return 0;

}

if(fabs(((y2-y1)/(x2-x1))-((y3-y1)/(x3-x1)))<0.0001)

printf("\npunkty sa wspolliniowe");

else

printf("\npunkty nie sa wspolliniowe");

return 0;

}

Zbadany zostaje najpierw najprostszy przypadek, gdy wszystkie punkty leżą na jednym x oraz przypadek, gdy dwa punkty leżą na jednym x – eliminuje to ewentualność dzielenia przez zero w dalszej części programu dla bardziej skomplikowanych ustawień punktów względem siebie. „fabs(((y2-y1)/(x2-x1))-((y3-y1)/(x3-x1)))<0.0001)” jest na wypadek odchyłów w obliczeniach przez operowanie raz w liczbach dziesiętnych, a raz w binarnych.

Zadanie nr 16.

Szyfrowanie i deszyfrowanie kodem Cezara dla zadanej odległości liter w alfabecie.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define n 1000

int main()

{

char cezar[n];

int odleglosc,wybor,i;

printf("Wpisz wiadomosc:\n");

fgets(cezar, n, stdin);

strupr(cezar);

printf("\nWpisz odleglosc liter: ");

scanf("%d",&odleglosc);

do

{

system("cls");

printf("1-zaszyfrowanie\n2-deszyfrowanie\nWybor: ");

scanf("%d",&wybor);

}while(wybor!=1 && wybor!=2);

switch(wybor)

{

case 1:

{

for(i=0;i<n;i++)

{

if(cezar[i]<=90 && cezar[i]>=65)

{

cezar[i]+=odleglosc;

if(cezar[i]>90)

cezar[i]-=26;

if(cezar[i]<65)

cezar[i]+=26;

}

}

break;

}

case 2:

{

for(i=0;i<n;i++)

{

if(cezar[i]<=90 && cezar[i]>=65)

{

cezar[i]-=odleglosc;

if(cezar[i]>90)

cezar[i]-=26;

if(cezar[i]<65)

cezar[i]+=26;

}

}

break;

}

}

printf("Wynik:\n\n%s",cezar);

return 0;

}

Funkcja fgets pozwala wczytywać dane razem ze spacjami, co nie jest możliwe przy scanf. Strupr zamienia wszystkie małe litery na wielkie, co ułatwia dalsze działanie na tekście. Do while nie pozwala wybrać innej opcji niż przewidziane. Konstrukcja switch zapewnia odpowiednie działanie według wybranej opcji. Między 65 a 90 znajdują się wielkie litery alfabetu łacińskiego w ASCII przez co dodając podaną liczbę przesuwamy się o tyle w prawo w alfabecie, a odejmując w lewo. W przypadku przekroczenia alfabetu poza „a” lub „z” odpowiednio dodajemy/odejmujemy 26 – liczbę liter w alfabecie łacińskim.