



Facultatea de Inginerie Electrică



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA



EUROPEAN UNIVERSITY
OF TECHNOLOGY

Facultatea de Inginerie Electrică

PCLP 2

Programarea calculatoarelor si limbaje de programare 2

PCLP2

An I semestrul II



"Coding is easy when you C it in action."

Cap. 12

Aplicatii C/C++ in inginerie

12.1. Polinoame, derivare polinoame, etc

12.2. Descompunere in serie Taylor a functiilor trigonometrice

12.3. Aplicatii cu valori medii, efective, media valorilor absolute a deviatiilor , calcul erori

12.4. Aplicatii cu integrale

12. Aplicatii C/C++ in inginerie

Aplicatii matematice in inginerie in C/C++

- ❑ **calcul expresii matematice** : constante, expresii,operatii cu siruri, matrici, etc.
- ❑ **operatii cu polinoame:** derivari polinoame, schema lui Horner, etc
- ❑ **valori medii , valori efective, valori medii absolute, valoarea medie patratica, media valorii absolute a deviatiilor, eroare medie patratica** (deviatiia standard)
- ❑ **calcul functii utilizand descompuneri in serii** : functii trigonometrice prin descompuneri in serie Taylor
- ❑ **calcul integrale** definite, prin aproximari utilizand metode diverse: metoda dreptunghiurilor, metoda trapezelor, metoda Simpson, metoda trecerii functiilor de sub integrale la limita sub forma de sume Riemann, integrale multidimensionale, etc
- ❑ **rezolvarea ecuatiilor algebrice utilizand metode numerice:** Metoda bisectiei, aproximatiilor succesive, Newton, secantei, Birge-Vieta
- ❑ **rezolvarea sistemelor de ecuatii algebrice liniare** (circuite electrice) utilizand metode numerice: Metoda de eliminare Gauss, Gauss –Jordan, Choleski, iterativa Jacobi, etc.

12. Aplicatii C/C++ in inginerie

Constante matematice

Symbol	Expression	Value
M_E	e	2.71828182845904523536
M_LOG2E	$\log_2(e)$	1.44269504088896340736
M_LOG10E	$\log_{10}(e)$	0.434294481903251827651
M_LN2	$\ln(2)$	0.693147180559945309417
M_LN10	$\ln(10)$	2.30258509299404568402
M_PI	pi	3.14159265358979323846
M_PI_2	$\pi/2$	1.57079632679489661923
M_PI_4	$\pi/4$	0.785398163397448309616
M_1_PI	$1/\pi$	0.318309886183790671538
M_2_PI	$2/\pi$	0.636619772367581343076
M_2_SQRTPI	$2/\sqrt{\pi}$	1.12837916709551257390
M_SQRT2	$\sqrt{2}$	1.41421356237309504880
M_SQRT1_2	$1/\sqrt{2}$	0.707106781186547524401

Constantele matematice nu sunt definite in Standard C/C++.

Pentru a le utiliza exista 2 variante:

- ❑ le definim ca si constante cu `#define` sau `const`,
- ❑ introducem `#define _USE_MATH_DEFINES` si `#include <math.h>` in C, sau `#include <cmath>` in C++

```
#define _USE_MATH_DEFINES // for C++
#include <cmath>
```

```
#define _USE_MATH_DEFINES // for C
#include <math.h>
```

12. Aplicatii C/C++ in inginerie

Constante matematice

EXAMPLE

Ex.1. Se afiseaza constante matematice

```
#include <stdio.h>
#include <stdlib.h>
#define _USE_MATH_DEFINES // for C
#include <math.h>
int main()
{ printf("pi=%lf\n",M_PI);
  printf("pi/2=%lf\n",M_PI_2);
  printf("pi/4=%lf\n",M_PI_4);
  printf("1/pi=%lf\n",M_1_PI);
  printf("2/pi=%lf\n",M_2_PI);
  printf("e=%lf\n",M_E);
  printf("log2(e)=%lf\n",M_LOG2E);
  return 0;}
```

```
pi=3.141593
pi/2=1.570796
pi/4=0.785398
1/pi=0.318310
2/pi=0.636620
e=2.718282
log2(e)=1.442695
```

12. Aplicații C/C++ în inginerie

Polinoame

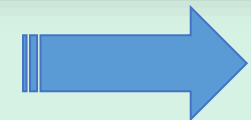
EXEMPLE

Ex.1. Se consideră un polinom de gradul n cu coeficienții numere reale introduse de la tastatură: $P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. Să se scrie programul care determină și afișează elementele unei matrici cu $n+1$ coloane și $n+1$ linii care conține pe linii coeficienții polinomului dat precum și coeficienții polinoamelor care rezultă în urma derivării succesive ale polinomului inițial (până când se ajunge la un polinom de gradul 0).

```
#include<stdio.h>
int i,j,n; float a[20][20];
void citire()
{printf("Gradul polinomului :");scanf("%d",&n);
printf("Coeficientii polinomului (incepand cu termenul liber):\n");
for(j=1;j<=n+1;j++)
    {if(j==1)printf("Termenul liber :");
    else printf("Coeficientul lui x^%d :",j-1); scanf("%f",&a[1][j]); }}
void deriv()
{for(i=2;i<=n+1;i++)
    for(j=1;j<=n+1;j++)
        a[i][j]=a[i-1][j+1]*j;}
//formarea matricii cu termenii polinomului derivat
```

```
Gradul polinomului :5
Coeficientii polinomului (incepand cu termenul liber):
Termenul liber :1
Coeficientul lui x^1:2
Coeficientul lui x^2:3
Coeficientul lui x^3:4
Coeficientul lui x^4:5
Coeficientul lui x^5:6
```

1	2	3	4	5	6
2	6	12	20	30	0
6	24	60	120	0	0
24	120	360	0	0	0
120	720	0	0	0	0
720	0	0	0	0	0



12. Aplicatii C/C++ in inginerie

Polinoame

EXAMPLE

Continuare:

```
void afis2()
//afisarea coeficientilor polinomului derivat
{ printf("\nDerivarea polinomului :");
  for(i=1;i<=n+1;i++)
  {printf("\n");
   printf("P(%d) = ",i-1);
    for(j=1;j<=n+1;j++)
      if(a[i][j]= =0) printf("");
      else if(j-1= =0)printf(" + %.f",a[i][j]);
      else if(j-1= =1) printf(" + %.fx",a[i][j]);
      else if(a[i][j]= =1) printf(" + x^%d",j-1);
      else printf(" + %.fx^%d",a[i][j],j-1); }

  printf("\nP(%d) = 0",i-1);}
void afis()
// afisarea matricii formate din coeficientii polinomului derivat
{printf("\n");
  for(i=1;i<=n+1;i++) {for(j=1;j<=n+1;j++) printf("%4.f ",a[i][j]);printf("\n");}
int main()
{citire(); deriv(); afis(); afis2();return 0;}
```

```
Gradul polinomului :5
Coeficientii polinomului (incepand cu termenul liber):
Termenul liber :1
Coeficientul lui x^1:2
Coeficientul lui x^2:3
Coeficientul lui x^3:4
Coeficientul lui x^4:5
Coeficientul lui x^5:6
```

1	2	3	4	5	6
2	6	12	20	30	0
6	24	60	120	0	0
24	120	360	0	0	0
120	720	0	0	0	0
720	0	0	0	0	0

```
Derivarea polinomului :
P(0)= + 1+2x+3x^2+4x^3+5x^4+6x^5
P(1)= + 2+6x+12x^2+20x^3+30x^4
P(2)= + 6+24x+60x^2+120x^3
P(3)= + 24+120x+360x^2
P(4)= + 120+720x
P(5)= + 720
P(6) = 0
```

12. Aplicații C/C++ în inginerie

Polinoame

EXEMPLE

Ex.3. Programul determină și afișează matricea care are pe linii coeficienții tuturor polinoamelor Cebășev, pentru gradul n întreg citit de la tastatură. Polinoamele lui Cebășev se definesc astfel [20]: $T_n(x) = \cos(n \cdot \arccos x)$, $|x| \leq 1$. Astfel $T_0(x) = 1$, $T_1(x) = x$, iar relația de recurență între polinoamele lui Cebășev este: $T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x)$. Relația de recurență dintre coeficienții polinoamelor lui Cebășev este:

$$t_{i,j} = 2t_{i-1,j-1} - t_{i-2,j}, \quad i = 2, 3, \dots, n; \quad 1 \leq j \leq i.$$

Se ține cont de asemenea de relația:

$$t_{i,0} = -t_{i-2,0}, \quad i = 2, 4, 6, 8, \dots$$

```
#include <stdio.h>
#include <math.h>
int main (void)
{int t[20][20], i,j,n;
printf("n="); scanf("%d", &n);
for (i=0;i<n;i++) for (j=0;j<n;j++) t[i][j]=0; t[0][0]=t[1][1]=1;
for (i=2;i<n;i+=2) t[i][0]=-t[i-2][0];
for (i=2;i<n;i++) for (j=1;j<=i;j++) t[i][j]=2*t[i-1][j-1]-t[i-2][j];
printf("Tn(x)=cos[n*arccos(x)]\n"); printf("Formula de recurenta ptr.polinoamele Chebyshev:\n");
printf("T0(x)=1\ t 1(x)=x\n");printf("Ti(x)=2*x*Ti-1(x)-Ti-2(x),i>1\n"); printf("Formula de recurenta pentru coeficientii polin.Chebyshev\n");
printf("t[i][j]=2*t[i-1][j-1]-t[i-2][j], i>=j>0\n");
printf("\n\tCoeficientii polinomului Chebyshev\n"); printf("*****\n");
printf("grad ");
for (i=0;i<n;i++) printf(" C%d ",i); printf("\n\n");
for(i=0;i<n;i++) { printf(" %2d ",i); for(j=0;j<n;j++) printf("%5d",t[i][j]); printf("\n");} return 0;}
```

```
n=10
Tn(x)=cos[n*arccos(x)]
Formula de recurenta ptr.polinoamele Chebyshev:
T0(x)=1 T1(x)=x
Ti(x)=2*x*Ti-1(x)-Ti-2(x),i>1
Formula de recurenta pentru coeficientii polin.Chebyshev
t[i][j]=2*t[i-1][j-1]-t[i-2][j], i>=j>0
```

Coeficientii polinomului Chebyshev										

grad	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	-1	0	2	0	0	0	0	0	0	0
3	0	-3	0	4	0	0	0	0	0	0
4	1	0	-8	0	8	0	0	0	0	0
5	0	5	0	-20	0	16	0	0	0	0
6	-1	0	18	0	-48	0	32	0	0	0
7	0	-7	0	56	0	-112	0	64	0	0
8	1	0	-32	0	160	0	-256	0	128	0
9	0	9	0	-120	0	432	0	-576	0	256

12. Aplicații C/C++ în inginerie

Aplicații matematice în inginerie: valori medii, erori

Ex.5. Să se scrie un program care citește valorile a n date a_i , $i=1, n$ de la tastatură, calculează și afișează valorile medii și absolute, valoarea medie patrată și valoarea efectivă, media valorilor absolute a deviațiilor, eroarea medie pătratică [20].

- Valoarea medie (aritmetică):

$$v_m = \frac{\sum_{i=1}^n a_i}{n}$$

- Valoarea efectivă:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (v_m - a_i)^2} = \sqrt{v_e^2 - v_m^2}$$

- Valoarea medie absolută:

$$v_A = \frac{\sum_{i=1}^n |a_i|}{n} \quad v_e = \sqrt{\frac{\sum_{i=1}^n a_i^2}{n}}$$

- Valoarea medie patrată :

$$v_p = v^2 = \frac{\sum_{i=1}^n a_i^2}{n}$$

- Media valorilor absolute a deviațiilor:

$$v_{MAD} = \frac{\sum_{i=1}^n |a_i - v_m|}{n}$$

- Eroarea medie patrată (deviația standard):

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (v_m - a_i)^2} = \sqrt{v_e^2 - v_m^2}$$

12. Aplicatii C/C++ in inginerie

Aplicatii matematice in inginerie: valori medii, erori

EXAMPLE

```
#include <stdio.h>
#include <math.h>
int main(void)
{int i,n;double a[100],er,s1=0,s2=0,s3=0,vA,vm,vp,ve,vMAD;
printf("Introduceti nr. de date de intrare:");scanf("%d",&n);
for (i=0;i<n;i++) { printf("a[%2d]=", i);scanf("%lf", &a[i]);}
for (i=0;i<n;i++) { s1+=a[i];s2+=fabs(a[i]); s3+=a[i]*a[i]; }
vm=s1/n; vA=s2/n; vp=s3/n;ve=sqrt(vp);
printf(" valoarea medie aritmetica:\nvm=%lf\t vm=%le \n",vm,vm);
printf(" valoarea medie absoluta:\nvA=%lf\t vA=%le \n", vA, vA);
printf(" valoarea medie patratica:\nvp=: %lf\t vp=%le \n", vp, vp);
printf(" valoarea efectiva:\nve=%lf\t ve=%le \n", ve, ve);
for (i=0,s1=0;i<n;i++) {s1+=fabs(a[i]-vm);}
vMAD=s1/n; er=sqrt(ve*ve-vm*vm);
printf(" media val.abs.a deviatiilor:\n");
printf("vMAD=%lf\t vMAD=%le \n", vMAD, vMAD);
printf(" eroarea patratica medie:\ner=%lf\t er=%le \n", er, er);
return 0;}
```

```
Introduceti nr. de date de intrare:6
a[ 0]=1.45
a[ 1]=6.35
a[ 2]=3.45
a[ 3]=2.88
a[ 4]=1.21
a[ 5]=6.22
valoarea medie aritmetica:
vm=3.593333      vm=3.593333e+000
valoarea medie absoluta:
vA=3.593333     vA=3.593333e+000
valoarea medie patratica:
vp=: 17.129067   vp=1.712907e+001
valoarea efectiva:
ve=4.138728     ve=4.138728e+000
media val.abs.a deviatiilor:
vMAD=1.794444   vMAD=1.794444e+000
eroarea patratica medie:
er=2.053539     er=2.053539e+000
```

12. Aplicații C/C++ în inginerie

Descompuneri în serii (Taylor)

Ex.1. Să se scrie programul care determină și afișează valoarea funcției sinus într-un punct dat x , utilizând primii n termeni din dezvoltarea în serie de mai jos și realizând comparația între rezultatele obținute astfel cu cele obținute prin apelarea directă a funcției `sin()` din biblioteca `<math.h>`:

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{x^{2i+1}}{(2i+1)!}$$

```
x=1.5
n=10
Val functiei sin(x) calculata
cu descompunerea in serie este:
sin1(1.500000)=0.99749498660405456000

val functiei sin(x) calculata
cu functia sin din <math.h> este:
sin2(1.500000)=0.99749498660405445000
```

EXEMPLE

```
#include <stdio.h>
#include <math.h>
double fact(int j)
{double t; int k; t=1.;
for (k=1;k<=j;k++) t=t*k; return t;}
int main(void)
{int i,n; double x,fx;
printf("X="); scanf("%lf",&x); printf("n="); scanf("%d",&n);
fx=x;
for (i=1;i<=n;i++) fx+=pow(-1,i)*pow(x,(2*i+1))/fact(2*i+1);
printf("Val funcției sin(x) calculată\ncu descompunerea în
serie este:\n");
printf("sin1(%lf)=%5.20lf\n", x, fx);
printf("\nval funcției sin(x) calculată\ncu funcția sin din
<math.h> este:\n");
printf("sin2(%lf)=%5.20lf\n", x ,sin(x)); return 0;}
```

12. Aplicații C/C++ în inginerie

Integrale

Ex.1. Programul calculează integrala definită astfel:

$$v = \int_a^b x^n dx, \quad 0 < a \leq b, \quad n \in \mathbb{Z}$$

pentru valori ale lui a,b care respectă condițiile de mai sus.

Rezolvare:

Valoarea exactă a integralei se va calcula conform relației :

$$v = \begin{cases} \ln \frac{b}{a}, & n = -1 \\ \frac{b^{n+1} - a^{n+1}}{n+1}, & n \neq -1 \end{cases}$$

```
Introduceti n intreg, n=3
Introduceti a diferit de 0, a=1
Introduceti b diferit de 0, b>=a>0,b=2
integrala de la 1 la 2 din x
la puterea 3 dx =3.750000
```

EXEMPLE

```
#include <stdio.h>
#include <math.h>
int main(void)
{int n; double a,b,v;
printf("Introduceti n intreg, n=");scanf("%d", &n);
printf("Introduceti a diferit de 0, a=");scanf("%lf", &a);
printf("Introduceti b diferit de 0, b>=a>0,b=");scanf("%lf", &b);
if (n== -1) v=log(b/a) ;
else v=(pow(b,n+1)-pow(a,n+1))/(n+1);
printf("integrala de la %2.lf la %2.lf din x\n", a,b);
printf("la puterea %2d dx =%lf\n", n,v);return 0;}
```

12. Aplicații C/C++ în inginerie

EXEMPLE

Integrale

Ex.3. Programul realizează calculul aproximativ al integralei definite prin:

$$I = \int_a^b f(x) dx$$

utilizând trei metode diferite: metoda dreptunghiurilor, metoda trapezelor și metoda lui Simpson [3], [4], [20].

Rezolvare:

- Metoda dreptunghiurilor: $\int_a^b f(x) dx \approx h \sum_{i=1}^n f(a - \frac{h}{2} + ih)$

- Metoda trapezelor:

$$\int_a^b f(x) dx \approx \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^n f(a + ih) - f(b) \right]$$

- Metoda lui Simpson:

$$\int_a^b f(x) dx \approx \frac{h}{6} \left[f(a) + 4 \sum_{i=1}^n f(a - \frac{h}{2} + ih) + 2 \sum_{i=1}^n f(a + ih) - f(b) \right]$$

unde $h = \frac{b-a}{n}$ este mărimea subintervalelor în care se face împărțirea intervalului $[a,b]$.

Constanta lui Catalan: $G = \int_0^1 \frac{\operatorname{arctg} x}{x} dx = 0.915965954$

În toate formulele se utilizează $f(a)$ unde $a=0$, astfel încât $f(x)$ se determină cu relația:

$$f(x) = \begin{cases} 1 & , x = 0 \\ \frac{\operatorname{arctg} x}{x} & , x \neq 0 \end{cases}$$

```
#include <stdio.h>
#include <math.h>
double f(double x);
int main(void)
{int i,n;
double a=0.,b=1,G=0.915965594,h,sc,sm,MD,MT,MS;
printf("\n\t Valoarea integralei G=%12.9lf\n", G);
printf("\n\tMet.dreptungh.\tMet.Trapezelor\tMet.Simpson\n");
printf("*****\n");
for (n=1;n<=20;n++) {h=(b-a)/n;
for(sm=0.,sc=0.,i=1;i<=n;i++) {sm+=f(a-(1/2)*h+i*h);
sc+=f(a+i*h);}
MD=h*sm; MT=0.5*h*(f(a)+2*sc-f(b));
MS=(h/6)*(f(a)+4*sm+2*sc-f(b));
printf("%2d\t%12.10lf\t%12.10lf\t%12.10lf\n",n,MD,MT,MS); }
return 0;}
double f(double x) {if (x) return atan(x)/x;
else return 1.0;}
```

12. Aplicatii C/C++ in inginerie

Integrale

Valoarea integralei $G=0.915965594$

	Met.dreptungh.	Met.Trapezelor	Met.Simpson
1	0.7853981634	0.8926990817	0.8211651362
2	0.8563466907	0.9099971499	0.8742301771
3	0.8775512440	0.9133182167	0.8894735682
4	0.8876523781	0.9144776076	0.8965941213
5	0.8935534501	0.9150136337	0.9007068446
6	0.8974211576	0.9153046440	0.9033823198
7	0.9001513549	0.9154800576	0.9052609225
8	0.9021812700	0.9155938848	0.9066521416
9	0.9037495897	0.9156719140	0.9077236978
10	0.9049976307	0.9157277225	0.9085743280
11	0.9060143828	0.9157690117	0.9092659258
12	0.9068586708	0.9158004140	0.9098392518
13	0.9075709345	0.9158248513	0.9103222401
14	0.9081798897	0.9158442410	0.9107346735
15	0.9087064887	0.9158598833	0.9110909536
16	0.9091663776	0.9158726850	0.9114018134
17	0.9095714759	0.9158832947	0.9116754155
18	0.9099310234	0.9158921855	0.9119180774
19	0.9102522930	0.9158997097	0.9121347652
20	0.9105410878	0.9159061337	0.9123294365

12. Aplicații C/C++ în inginerie

Integrale

Ex.4. Programul realizează calculul expresiei $E(a)$ pentru diferite valori ale lui a și n , unde :

$$E(a) = \frac{2}{\sqrt{\pi}} \int_0^a e^{-x^2} dx$$

Se știe că $E(a) \rightarrow 1$ când $a \rightarrow \infty$. Integrala definită se calculează aproximativ prin intermediul formulei generalizate a trapezelor [3], [4], [20]:

$$\int_a^b f(x) dx \approx \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^n f(a+ih) - f(b) \right], \quad h = \frac{b-a}{n}$$

EXEMPLE

```
#include <stdio.h>
#include <math.h>
int main ()
{int i,n;
double a, b, h, r,ra, rb,rc;
printf("a="); scanf("%lf",&a);
printf("n=");scanf("%d",&n);
b=0.; h=(b-a)/n;
ra=h/2*exp(-(pow(a,2)));
rc=h/2*exp(-(pow(b,2)));
rb=0.;
for (i=1;i<=n;i++) { rb+=exp(-(pow(a-i*h,2)));}
rb*=h;
r=(2/sqrt(3.14))*(ra+rb-rc);
printf("E(%.2lf)=%12lf\n", a,r);
return 0;}
```

a=100

n=10

E(100.00)= 5.643326

12. Aplicatii C/C++ in inginerie

Integrale

Ex.6. Programul realizează rezolvarea integralelor de mai jos prin trecerea expresiilor funcțiilor de sub integrale, la limită sub forma unor sume Riemann [3], [4], [20]:

$$I_1 = \int_0^1 \frac{1}{x+1} dx = \lim_{n \rightarrow \infty} \left(\frac{1}{n+1} + \frac{1}{n+2} + \dots + \frac{1}{2n} \right) = \ln 2 = 0.6931471805$$

$$I_2 = \int_0^1 \frac{1}{\sqrt{4-x^2}} dx = \lim_{n \rightarrow \infty} \left(\frac{1}{\sqrt{4n^2-1^2}} + \frac{1}{\sqrt{4n^2-2^2}} + \dots + \frac{1}{\sqrt{4n^2-n^2}} \right) = \pi/6 = 0.5235987755$$

$$I_3 = \int_0^1 x^4 dx = \lim_{n \rightarrow \infty} \frac{1}{n^5} (1^4 + 2^4 + \dots + n^4) = \frac{1}{5} = 0.2$$

$$I_4 = \int_0^1 x^5 dx = \lim_{n \rightarrow \infty} \frac{1}{n^6} (1^5 + 2^5 + \dots + n^5) = \frac{1}{6} = 0.166666$$

$$I_6 = \int_0^1 \frac{1}{1+x^2} dx = \lim_{n \rightarrow \infty} n \left[\frac{1}{n^2+1^2} + \frac{1}{n^2+2^2} + \dots + \frac{1}{n^2+n^2} \right] = \pi/4 = 0.7853981633$$

$$I_7 = \int_0^{\pi} \sin x dx = \lim_{n \rightarrow \infty} \frac{\pi}{n} \left(\sin \frac{\pi}{n} + \sin \frac{2\pi}{n} + \dots + \sin \frac{n\pi}{n} \right) = 2.0$$

$$I_8 = \int_0^1 \frac{1}{(1+x)^2} dx = \lim_{n \rightarrow \infty} n \left[\frac{1}{(n+1)^2} + \frac{1}{(n+2)^2} + \dots + \frac{1}{(2n)^2} \right] = \frac{1}{2} = 0.5$$

$$I_9 = \int_0^1 \frac{1}{\sqrt{2x+5}} dx = \lim_{n \rightarrow \infty} \frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{2+5n}} + \frac{1}{\sqrt{4+5n}} + \dots + \frac{1}{\sqrt{2n+5n}} \right) = \sqrt{7} - \sqrt{5} = 0.4096833335$$

$$I_{10} = \int_0^1 \sqrt{1-x^2} dx = \lim_{n \rightarrow \infty} \frac{1}{n^2} \left(\sqrt{n^2-1^2} + \dots + \sqrt{n^2-n^2} \right) = \frac{\pi}{4}$$

12. Aplicatii C/C++ in inginerie

Integrale

EXAMPLE

```
//Programul ex6.c
#include <stdio.h>
#include <math.h>
void main()
{int i,j,n;
double part=0., e=2.718282, pi=3.14159;
double I1exact=0.6931471805; double I2exact=0.5235987755;
double I3exact=0.2; double I4exact=0.1666666666;
double I6exact=pi/4; double I7exact=2.;
double I8exact=1./2.;double I9exact=0.4096833335;
double I10exact=pi/4.;
printf("aproximarea integralelor prin sume Riemann\n"); printf("n=");scanf("%d", &n);
//integrala 1
for (i=1; i<=n;i++)
{      part=0.      ;
      for (j=1;j<=i;j++) part+=1./(i+j);}
printf("\n n=%d, I1aprox=%10lf, I1exact=%10lf\n eroarea=%10lf\n", n, part, I1exact,
(I1exact-part));// linie continuata de pe randul precedent
```

12. Aplicatii C/C++ in inginerie

Integrale

EXAMPLE

```
//integrala 2
for (i=1; i<=n;i++)
{
    part=0.    ; for (j=1;j<=i;j++) part+=1./sqrt(4*pow(i,2)-pow(j,2));}
printf("\n n=%d, I2aprox=%10lf, I2exact=%10lf\n eroarea=%10lf\n", n,
part,I2exact,(I2exact-part));
//integrala 3
for (i=1; i<=n;i++)
{
    part=0.    ; for (j=1;j<=i;j++) {part+=pow(j,4);}
part*=1/pow(i,5);}
printf("\n n=%d, I3aprox=%10lf, I3exact=%10lf\n eroarea=%10lf\n", n, part,
I3exact,(I3exact-part));// linie continuata de pe randul precedent !
//integrala 4
for (i=1; i<=n;i++)
{
    part=0.    ; for (j=1;j<=i;j++) {part+=pow(j,5);}
part*=1/pow(i,6);}
printf("\n n=%d, I4aprox=%10lf, I4exact=%10lf\n eroarea=%10lf\n", n,
part,I4exact,(I4exact-part)); // linie continuata de pe randul precedent !
...
```

```
n=1000, I1aprox= 0.692897, I1exact= 0.693147
eroarea= 0.000250

n=1000, I2aprox= 0.523637, I2exact= 0.523599
eroarea= -0.000039

n=1000, I3aprox= 0.200500, I3exact= 0.200000
eroarea= -0.000500

n=1000, I4aprox= 0.167167, I4exact= 0.166667
eroarea= -0.000500

n=1000, I6aprox= 0.785148, I6exact= 0.785397
eroarea= 0.000249

n=1000, I7aprox= 1.999998, I7exact= 2.000000
eroarea= 0.000002

n=1000, I8aprox= 0.499625, I8exact= 0.500000
eroarea= 0.000375

n=1000, I9aprox= 0.409649, I9exact= 0.409683
eroarea= 0.000035

n=1000, I10aprox= 0.785499, I10exact= 0.785397
eroarea= -0.000102
```