**Numele şi prenumele: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Anul I II III, Semianul A B E, Grupa 1 2 3 4 5 6 7 (încercuiţi) Altceva: \_\_\_**

**Introducere în programare H**

**1. Care din următoarele funcții “diamant” nu desenează o figură ca cea din stânga, dacă este inclusă în programul din dreapta?**

|  |  |
| --- | --- |
| 1. funcția 1 2. funcția 2 3. funcția 3 4. funcția 4 | #include <iostream>  #include <graphics.h>  #include <winbgim.h>  void patrat(int x,int y, int L)  {  int L2=L/2;  line(x-L2,y-L2,x+L2,y-L2);  line(x+L2,y-L2,x+L2,y+L2);  line(x+L2,y+L2,x-L2,y+L2);  line(x-L2,y+L2,x-L2,y-L2);  }  int main()  {  initwindow(800,600);  **diamant(400,300,200);**  getch(); closegraph();  return 0;  } |

|  |  |
| --- | --- |
| // functia 1  void diamant(int x, int y, int L)  {  if (L>10) {  patrat(x,y,L);  diamant(x-L,y,L/2);  diamant(x+L,y,L/2); }  } | // functia 2  void diamant(int x, int y, int L)  {  if (L>10) {  patrat(x,y,L);  diamant(x+L/2,y,L/4);  diamant(x-L/2,y,L/4); }  } |
| // functia 3  void diamant(int x, int y, int L)  {  if (L>10) {  patrat(x,y,L);  diamant(x-L/2,y,L/2);  diamant(x+L/2,y,L/2); }  } | // functia 4  void diamant(int x, int y, int L)  {  if (L>10) {  patrat(x,y,L);  diamant(x,y+L/2,L/2);  diamant(x,y-L/2,L/2); }  } |

**2. Indicaţi care dintre expresiile C/C++ de mai jos are valoarea 1 dacă şi numai dacă numărul memorat în variabila întreagă x aparţine mulțimii {-3,-1} U {1, 3}.**

a) x>=-3 && x<=-1 && x>=1 && x<=3 b)!(x<-3 || x>-1) || !(x<1 || x>3)

c) x>=-3 || x<=-1 || x>=1 || x<=3 d) !(x<-3 && x>3 && x>-1 || x<1)

e) *niciuna din variantele a-d*

**3. Ce afișează următorul program?**

#include <stdio.h>

int main() {

enum culoare {alb, negru=-3, verde, albastru=2, rosu};

enum culoare x=negru; enum culoare y=albastru; int z=x+y;

printf("%d %d %d\n",x,y,z); return 0; }

a) 3 2 -1 b) -3 2 1 c) -3 2 -1 d) BLACK GREEN BLUE e) *altceva*

f) *program greșit* g) *niciuna din variantele a-f*

**4. Fie programul următor. Care din răspunsurile următoare este/sunt incorect(e) ?**

#include <iostream>

using namespace std;

int b,c,d; int a[]={3,7,2,5,6};

void test(int a, int &b, int \*c) {

for(int i=1;i<4;i++) \*(c+i)=a\*b++; a\*=\*(c+2); b=a\*\*c; d=a++\*\*c;

cout<<a<<' '<<b<<' '\*c<<' '<<d; }

int main() {

b=1; c=2; int d=3; test(c,b,a); cout<<\*(a+3)<<' '<<c<<' '<<d<<endl;

}

a) programul afișează 9 24 3 246 2 3 c) apelul funcției „test” este incorect din punct de vedere sintactic

b) programul afișează 9 24 3 246 9 3 d) funcția „test” este corectă din punct de vedere sintactic

**5. Se consideră funcţiile recursive C1 şi C2, definite mai jos.**

|  |  |
| --- | --- |
| int C1 (int a, int b) {  if(a=b) return a;  else  if(a>b) return C1(a-b,b);  else return C1(a,b-a); } | int C2 (int a, int b) {  if (b=0) return a;  else return C2(b,a%b);  } |

La apel, returnează valoarea celui mai mare divizor comun al celor două numere naturale nenule primite ca parametri:

a) numai C1 b) numai C2 c) atât C1 cât şi C2 d) nici C1, nici C2 e) C1 sau C2

f) niciuna din variantele a-d

**6. Ce va afişa următorul program C++?**

#include <iostream>

using namespace std;

int a=1, b=4,c=1,d=1234;

int main() {

int a=2, b=5; cout<<(a>b?a+2:b>c?b+2:c>5?c+2:c\*=2)<<" ";

cout<< (--a>=++b-3 || b+3<=c-- && ++a-b>=c++)+--b\*c++; return 0;

}

a) 6 0 b) 10 35 c) 0 d) două numere, din care primul nu este nenul

e) 10 40 f) niciuna din variantele a-d // afiseaza 7 0

**7. Ce va afişa următorul program C++?**

#include <iostream>

using namespace std;

int main() {

struct test {char b; unsigned char f:2; unsigned int c:7; unsigned int d:2;

double e; unsigned long long a:5; };

cout<<sizeof(test); return 0; }

a) 40 b) 24 c) 32 d) 48 e) 16 f) niciuna din variantele a-e

**8. Ce se va afișa în urma executării programului următor?**

#include <iostream>

using namespace std;

int main()

{ int a=5,b=2,\*p=&a,\*q=&b; \*p=\*q+1; a+=\*q; b=\*p+b+1; p=&b;

q=&a; cout<<a<<' '<<b<<' '<<\*p<<' '<<\*q; }

a) 5 8 8 5 b) 5 5 8 8 c) 8 8 5 5 d) 8 5 8 5 e) niciuna din variantelea-d

**Toate subiectele sunt obligatorii. Consultarea oricărei resurse este interzisă. Timp de lucru: 45 minute. Fiecare subiect 2-8 are 1 punct, subiectul 1 are 2 puncte, iar 1 punct se acordă din oficiu.**