Problema 11:

Varianta 1: NU COMPILEAZA linia 10 da eroare

#include <iostream>

using namespace std;

template<class X>

X f(X x, X y)

{ return \*x-\*y; }

int f(double \*x, int y)

{ return \*(x+y); }

int main()

{ double a=10.7,\*b=new double(21);

cout<<f(a,b);

return 0; }

-a e double, b e double\* (sau subinteles) 0.1p

f(a,b)-nu gaseste potrivire

- fara conversie la non-template 0.1p

-fara conversie la template 0.1p

- nu se poate face conversie la non-template 0.1p

-varianta corecta 0.1p

Varianta 2: NU COMPILEAZA ca in operatorul << avem a.x si nu avem acces la x

#include <iostream>

using namespace std;

template<class X>

X f(X x, X y)

{ return \*x-\*y; }

int f(double \*x, int y)

{ return \*(x+y); }

int main()

{ double a=10.7,\*b=new double(21);

cout<<f(b,b);

return 0; }

- fara conversie la non-template 0.1p

-fara conversie la template 0.1p

-rezultat \*x -\*y de tip double

- nu face conversia catre double \* tipul intors de template 0.1

-varianta corecta 0.1p

Varianta 3: COMPILEAZA si afiseaza valoare nedetrminata

#include <iostream>

using namespace std;

template<class X>

X f(X x, X y)

{ return \*x-\*y; }

int f(double \*x, int y)

{ return \*(x+y); }

int main()

{ double a=10.7,\*b=new double(21);

cout<<f(b,a);

return 0; }

- fara conversie la non-template 0.1p

-fara conversie la template 0.1p

-cu conversie la non template: a la int 0.1p

-x+y ca pointer 0.1p

\_\*(x+y) zona adresata nedeterminata 0.1p

Varianta 4: COMPILEAZA si afiseaza valoare nedetrminata

#include <iostream>

using namespace std;

template<class X>

X f(X x, X y)

{ return \*x-\*y; }

int f(double \*x, int y)

{ return \*(x+y); }

int main()

{ double a=7.7,\*b=new double(21);

cout<<f(b,a);

return 0; }

- fara conversie la non-template 0.1p

-fara conversie la template 0.1p

-cu conversie la non template: a la int 0.1p

-x+y ca pointer 0.1p

\_\*(x+y) zona adresata nedeterminata 0.1p

Varianta 5: NU COMPILEAZA linia 10 da eroare

#include <iostream>

using namespace std;

template<class X>

X f(X x, X y)

{ return \*x-\*y; }

int f(double \*x, int y)

{ return \*(x+y); }

int main()

{ double a=7.7,\*b=new double(21);

cout<<f(a,a);

return 0; }

- fara conversie la non-template 0.1p

-fara conversie la template 0.1p

-\*x si \*y fara sens pt x si y double 0.2p

-varianta corecta 0.1p