1)#include <stdio.h>

#include <string.h>

void main() {

int \*aux = new int, i, \*\*tabela = new int\* [3];

for ( i = 0; i< 3 ; i++)

{ \*aux = i; tabela[i] = aux; }

for ( i = 0; i< 3 ; i++)

printf(" %d ", \*tabela[i]);

}

/\* Output-ul este :

2 2 2

\*/

2)#include <stdio.h>

#include <string.h>

void main() {

for ( int i = 1 ; i< 4 ; i++ ){

static int j=1; int k = 1;

printf("\n i=%d j=%d k=%d", i,j,k);

j++; k++;

}

}

/\* Output-ul este :

i=1 j=1 k=1

i=2 j=2 k=1

i=3 j=3 k=1

Ideea e ca desi static int j=1; j este static, tot isi schimba valoarea.

Sunt destept, nu? Ei, bine la examen NU am gandit asa.

\*/

3)#include <stdio.h>

#include <string.h>

void main() {

char buf [] = "%p%p%p%p%p%p%p%p";

printf(buf);

getchar();

}

/\* Output-ul este :

00000000000000007FFDA000CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

Ce dracu e asta? - Continutul stivei. Deci, printeaza continutul stivei.

\*/

a) Este eronata deoarece aapelul printf nu printeaza informatii de ...

b) afiseaza variabile sau adrese aflate pe stiva

c) este eronata ....

d) afiseaza variablile sau adrese din heap

e) Afiseaza var sau adrese din registrii

4)#

include <stdio.h>

#include <string.h>

void main() {

int lg = -10; char buf[100] = "Examen";

bool r = lg + strlen(buf) < sizeof(buf);

printf( r ? "Passed" : "Failed");

}

/\* Output-ul este :

Failed.

Explicatia este... in cartile de C++. Ceva de genul 'lg + strlen(buf)' se interpreteaza

ca unsigned de negativ ( -10 ) care este mult mai mare ca sizeof(buf).

Anyway, am facut ceva de genul la clasa.

Codul acolo era :

bool SafeCopy(char\* str, int size)

{

char buf[80];

if(size < sizeof(buf))

{

strcpy(buf, str); return true;

}

else return false;

}

4) Cu apel cu parametru negativ : printf( "\n %s", SafeCopy("Copiere sigura ???",-2) ? "Da": "Ba !!? );

Output : Ba !!

\*/

5)#include <stdio.h>

#include <string.h>

void main() {

int x = 7, y = 8, &ri = x;

printf("\n %d", ri);

ri = y; y = 10;

printf("\n %d", ri);

}

/\* Output-ul este :

7

8

\*/

6)#include <stdio.h>

#include <string.h>

void main() {

char \*pt = "01234567890";

puts(pt + 2);

getchar();

}

/\* Output-ul este :

234567890

\*/

7)#include <stdio.h>

#include <string.h>

void main() {

int count;

printf("1234567890123456%n7890", &count);

printf("\n %d", count);

getchar();

//printf(" Number displayed: %d ", count);

}

Ce afiseaza :

printf(" Number displayed: %d ", count); ?

/\*

Dar raspunsul pus de mine si confirmat si de cei cu care am vorbit

este ca afiseaza adresa lui count.

\*/

Amintiri din teorie

Subiectele de teorie su fost legate de :

8) o intrebare (SQL Injection) era care din 2 variante e mai sigura sau ceva de genul.

a) select \* from tabel where nume = " ana " ;

b) select \* from tabel where nume = ' " ana " ' " ;

Raspuns b) ca transforma la text input-ul. Trebuie retinut

ce scrie in cus : - Nu inlatura intotdeauna problema.

9) o intrebare era de genul :

Care din urmatoarele metode este mai periculoasa ?

Raspuns : Statement cu parametrii.

Intrebarea era legata de : CallableStatement, PrepareStatement si Statement.

10) o intrebare era legata de Watermarking (adica asta era raspunsul)

11) o intrebare era de genul : Ce reprezinta 'strong typing' ?

12) o intrebare era de genul : Ce reprezinta 'strong names' ?

a) politica de securitate

b) cultura cu care se asociaza

c) cheie publica

d) semnatura digitala

e) nr versiunii

13) Un atac care permite atacatorului doar sa impiedice derularea unui seviciu fara a castiga nimic altceva este cunoscut sub denumirea de:

a) Machine 'owning"

b) Service Linking

c) watermarketing

d) Obfuscation

e) DOS

14) Deoarece visual C.NEt se compileaza mai intai intr-un limbaj intermediar programul devine mai usor decompilabil?

a) Da, deoarece refacerea structurilor de programare devine mult mai simpla

b) Nu , deoarece limbajul intermediar este mult mai dificil de inteles de un atacator

c) Nu, deoarece limbajul intermediar este dependent de platform ade rulare.

d) Da, deoarece limbajul intermediar permite rularea pe o masina virtuala correct

e)Limbajul .Net este de tip script si nu se complileaza ci se interpreteaza

15) o intrebare era legata de termenul de 'managed code'

WTF is that? Ca eu nu am gasit nimic in curs despre asta.