

TEMA 6

PRINTSCREEN REZULTAT PROGRAM LUCRARE

```
Microsoft Visual Studio Debug Console

EFLAGS initial si cel modificat sunt egale.
Vendor ID: AuthenticAMD

Model Number: 1
Family Code: 15
Extended Mode: 1
Processor Type: 0
Extended Family: 8
Brand ID: 0

Flags FPU, VME, DE, PSE: 15

No. procesoare logice: 0

Info. despre cache:
0
0
0
0

C:\Users\Denis\Desktop\pmd\PMD\Debug\PMD.exe (process 13836) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

System

Control Panel > System and Security > System

Control Panel Home

- Device Manager
- Remote settings
- System protection
- Advanced system settings

View basic information about your computer

Windows edition

Windows 10 Home

© 2020 Microsoft Corporation. All rights reserved.

System

Processor:	AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx	2.00 GHz
Installed memory (RAM):	16.0 GB (14.9 GB usable)	
System type:	64-bit Operating System, x64-based processor	
Pen and Touch:	No Pen or Touch Input is available for this Display	

Computer name, domain, and workgroup settings

Computer name:	DESKTOP-IEAUO36
Full computer name:	DESKTOP-IEAUO36
Computer description:	
Workgroup:	WORKGROUP

[Change settings](#)

Windows activation

Windows is activated [Read the Microsoft Software License Terms](#)

Product ID: 00325-96555-46887-AAOEM

[Change product key](#)

See also

Security and Maintenance

RASPUNSURI INTREBARI

- 1. Folosind documentația Intel, explicați de ce se aplică următoarele operații pe biți, constând în deplasarea spre dreapta cu 4, 8, 12, 16, respectiv 20 de poziții, a variabilelor din porțiunea de cod de mai jos:**

```
vendorID[12] = '\0';  
  
cout << "Vendor ID: " << vendorID << "\n\n";  
  
modelNum >>= 4;  
FamilyCODE >>= 8;  
procTYPE >>= 12;  
ExtMODE >>= 16;  
extFam >>= 20;  
  
cout << "Model Number: " << modelNum << "\n";  
cout << "Family Code: " << FamilyCODE << "\n";  
cout << "Extended Mode: " << ExtMODE << "\n";  
cout << "Processor Type: " << procTYPE << "\n";  
cout << "Extended Family: " << extFam << "\n";
```

Deplasarea spre dreapta cu 4, 8, 12, 16, 20 de pozitii se foloseste pentru a afla informatia din pozitia registrului EAX. Ca exemplu putem da variabila FamilyCODE, care pentru a fi afisata corect trebuie deplasata cu 8 pozitii fiindca se afla de la pozitia 8 pana la pozitia 11 inclusiv, asemanator facandu-se si pentru celelalte variabile.

- 2. Explicați care este rolul instrucțiunilor: pushfd si pop eax.**

Rolul instructiunii PUSHFD este de a apela flags din reg. EFLAGS si de a salva continutul EFLAGS in stiva. Operatiile sunt pe 32 biti.

Rolul instructiunii POP EAX este de a copia din varful stivei si de a salva informatia in registrul EAX.

- 3. Folosind documentația Intel furnizată, scrieți care este registrul procesorului care va conține informațiile Extended Family și Extended Model în urma apelării instrucțiunii CPUID și care sunt pozițiile binare revendicate de fiecare dintre acestea.**

Registrul procesorului care va contine informatiile Extended Family si Extended Model in urma apelarii instructiunii CPUID este registrul EAX, adica Extended AX, iar

pozițiile binare pentru Extended Family sunt între 20 și 27, respectiv pentru Extended Model între 16 și 19.

4. Folosind documentația Intel furnizată, scrieți care este registrul procesorului care va conține informațiile APIC ID și Count în urma apelării instrucțiunii CUID și care sunt pozițiile binare revendicate de fiecare dintre acestea.

Registrul procesorului care va conține informațiile APIC ID și Count în urma apelării CUID este registrul EBX, iar pozițiile binare pentru Count sunt între 16 și 23, respectiv pentru APIC ID între 24 și 31.

5. Folosind documentația Intel furnizată, scrieți care ar trebui să fie conținutul binar al registrului EAX în urma apelului instrucțiunii CUID pentru procesoarele Intel 486 SX.

Conținutul registrului EAX în urma apelului instrucțiunii CUID pentru procesoarele Intel 486 SX sunt:

Stepping ID: xxxx

Model: 0010

Family: 0100

Processor type: 00

Extended Model: 0000

Extended Family: 00000000

6. Folosind documentația Intel furnizată, scrieți care ar trebui să fie conținutul binar al registrului EAX în urma apelului instrucțiunii CUID pentru procesoarele Intel Pentium Pro, precum și pentru procesoarele Intel Core i7.

Conținutul registrului EAX în urma apelului instrucțiunii CUID pentru Intel Pentium Pro este:

00000000	0000	00	0110	0001	xxxx (2)	Pentium Pro processor
----------	------	----	------	------	----------	-----------------------

Conținutul registrului EAX în urma apelului instrucțiunii CUID pentru Intel Core i7 este:

00000000	0001	00	0110	1010	xxxx (2)	Intel Core i7 processor and Intel Xeon processor. All processors are manufactured using the 45 nm process.
----------	------	----	------	------	----------	--

7. Explicați la ce este folosită variabila unsigned long int brandID din codul exemplu. Care biți, din care registru, vor fi salvați în această variabilă?

Variabila brandID este folosită pentru a salva informație despre procesor. Pozițiile bitilor folosiți sunt între 0 și 7, iar registrul folosit este EBX.

Processors that implement the Brand ID feature return the Brand ID in bits 7 through 0 of the EBX register when the CPUID instruction is executed with EAX=1 (see [Table 4-1](#)). Processors that do not support the feature return a value of 0 in EBX bits 7 through 0.