



Examen BP4



Universidad de Granada - Grado en Ingeniería Informática
Arquitectura de Computadores



Desconocido: **32081180** Llamas González, Alberto



Inicio: Hoy, lunes, 17:40:07

Final: Hoy, lunes, 17:59:10

Preguntas: 10

Respuestas
válidas:

Puntuación:

Nota:

1

Elección única

Dado el siguiente código y suponiendo el vector v inicializado, ¿qué opción es verdadera?

```
for (int i = 0; i < 1000; ++i)
{
    if ((v[i] % 3) == 0)
        foo(v[i]);
    else
        switch((v[i] % 3))
        {
            case 1: foo(v[i] + 2); break;
            case 2: foo(v[i] + 1); break;
        }
}
```

Usuario Profesores

- a) sólo el desenrollado de bucle puede servir para optimizar el código
- b) los valores contenidos en v no afectan a la velocidad de ejecución
- c) la ejecución finaliza antes si v no contiene ningún múltiplo de 3
- d) la ejecución finaliza antes si v contiene muchos

- ☐ a)

```
for (int k = 0; k < 100; ++k)
  for (int j = 0; j < 100; ++j)
    for (int i = 0; i < 100; ++i)
      a[i][j] += b[i][k] * c[k][j];
```
- ☐ b)

```
for (int j = 0; j < 100; ++j)
  for (int i = 0; i < 100; ++i)
    for (int k = 0; k < 100; ++k)
      a[i][j] += b[i][k] * c[k][j];
```
- ☐ c)

```
for (int i = 0; i < 100; ++i)
  for (int k = 0; k < 100; ++k)
    for (int j = 0; j < 100; ++j)
      a[i][j] += b[i][k] * c[k][j];
```
- ☐ d)

```
for (int i = 0; i < 100; ++i)
  for (int j = 0; j < 100; ++j)
    for (int k = 0; k < 100; ++k)
      a[i][j] += b[i][k] * c[k][j];
```

3

¿Cuál de las siguientes ventajas aporta la técnica de desenrollado de bucle?

Elección única

Usuario Profesores

- ☐ a) reduce el tamaño del código
- ☐ b) aumenta la velocidad de ejecución
- ☐ c) elimina todas las instrucciones de salto del bucle
- ☐ d) ninguna otra respuesta es correcta

4

¿Cuál de las siguientes versiones de una función que multiplica un entero por 6 cree que se obtendrá al compilar con optimización en espacio (-Os)?

Elección única

```
int f(int x)
{
    return x * 6;
}
```

Usuario Profesores

- ☐ a)

```
0x401106 <+0>: push    %rbp
0x401107 <+1>: mov     %rsp,%rbp
0x40110a <+4>: mov     %edi,-0x4(%rbp)
0x40110d <+7>: mov     -0x4(%rbp),%edx
0x401110 <+10>: mov     %edx,%eax
0x401112 <+12>: add     %eax,%eax
0x401114 <+14>: add     %edx,%eax
0x401116 <+16>: add     %eax,%eax
```

```

0x4005d0 <+0>: cmp    %esi, %edi
0x4005d2 <+2>: mov    %esi, %eax
0x4005d4 <+4>: cmovle %edi, %eax
0x4005d7 <+7>: retq

```

Usuario Profesores

- ☒ a)

```
int f(int a, int b) {
    if (a > b)
        return a;
    else
        return b;
}
```
- ☒ b)

```
int f(int a, int b) {
    if (a < b)
        return a;
    else
        return b;
}
```
- ☒ c) ninguna otra respuesta es correcta
- ☒ d)

```
int f(int a, int b, int c, int d) {
    if (a < b)
        return c;
    else
        return d;
}
```

6

¿Cuál de las siguientes afirmaciones es correcta?

Usuario Profesores


Elección única

- ☒ a) la optimización de código siempre debe realizarse en lenguaje ensamblador
- ☒ b) hay optimizaciones que son aplicables a cualquier procesador
- ☒ c) ninguna otra respuesta es correcta
- ☒ d) el proceso de optimización se debe realizar siempre al final del desarrollo de la aplicación

```

        sum += s.a[i];
        sum += s.b[i];
    }

```

 d) for (int i = 0; i < N; ++i)
 sum += s.a[i];
 for (int i = 0; i < N; ++i)
 sum += s.b[i];


8 ¿Cuál de las optimizaciones siguientes reduce el tiempo de ejecución del código que se muestra a continuación? (M y N son múltiplos de dos).
 Elección única


```


for (i=0; i<M; i++)
    for (j=0; j<N; j++){
        if ((j%2)==0)
            c[j] += a[j][i]+b[j][i];
        else
            c[j] += a[j][i]-b[j][i];
    }

```


Usuario Profesores


 a) for (i=0; i<M; i++)
 for (j=0; j<N; j++){
 c[j] += a[j][i]+b[j][i];
 c[j+1] += a[j+1][i]-b[j+1][i];
 }


•  b) for (i=0; i<M; i++)
 for (j=0; j<N; j+=2){
 c[j] += a[j][i]+b[j][i];
 c[j+1] += a[j+1][i]-b[j+1][i];
 }


 c) for (i=0; i<M; i+=2)
 f (j 0 j<N j+ 2){

Usuario Profesores

-  a)

```
for (i=0 ; i<n ; i++) {  
    *p = *p + a[i]*b[i];  
}
```
 -  b)

```
for (i=0 ; i<n ; i+=4) {  
    *p += a[i ]*b[i ];  
    *p += a[i+1]*b[i+1];  
    *p += a[i+2]*b[i+2];  
    *p += a[i+3]*b[i+3];  
}
```
 -  c)

```
for (i=0 ; i<n ; ++i) {  
    *p += a[i]*b[i];  
}
```
 -  d)

```
int tmp0=0, tmp1=0, tmp2=0, tmp3=0;  
for (i=0 ; i<n ; i+=4) {  
    tmp0 += a[i ]*b[i ];  
    tmp1 += a[i+1]*b[i+1];  
    tmp2 += a[i+2]*b[i+2];  
    tmp3 += a[i+3]*b[i+3];  
}  
*p = tmp0 + tmp1 + tmp2 + tmp3;
```
-