

A)

$$4 \cdot i \cdot M = 4 \cdot i + 16 \cdot i = 20i$$

$$M = \frac{20i}{4i} = 5$$

$$4 \cdot N = 28i$$

$$N = \frac{28i}{4i} = 7$$

B) 13 instructions.

C) 13 instructions.

D) 9 accesses

E)

$$0.5 \text{ inst/cycle} \Rightarrow 2 \text{ c/i}$$

$$0.8 \text{ inst/cycle} \Rightarrow 1.25 \text{ c/i}$$

$$9 \cdot 2 + 4 \cdot 1.25 = 23.5 \text{ cycles}$$

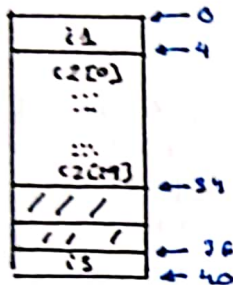
$$0.6 \text{ inst/c} = 1.6$$

$$0.9 \text{ inst/c} = 1.1$$

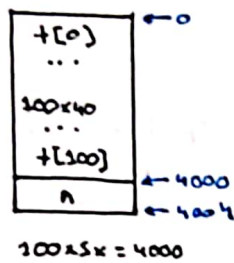
$$9 \cdot 1.6 + 4 \cdot 1.1 = 19.4 \text{ cycles}$$

1.19.-

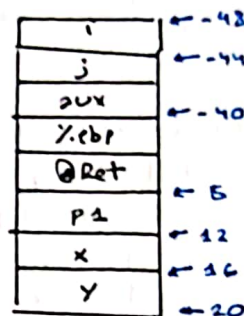
A)



5x=40



B)



C) return (\*x + aux[i]);

```
movl 12(%ebp), %eax
movl (%eax), %eax
addl -4(%ebp), %eax
```

D) aux[i] = F(2\*p2, i, j, y);

```
movl 8(%ebp), %eax
movl -44(%ebp), %ecx
imull $40, %ecx
addl %ecx, %eax
movl 16(%ebp), %ecx
pushl %ecx
pushl %eax
call F
addl $3, %esp
movl %eax, -16(%ebp)
```

E) i = j + y;

```
movl -44(%ebp), %eax // j
movl 16(%ebp), %ecx // y
imull %eax, %ecx // j*y
movl %ecx, 16(%ebp) // i
```

F) aux[c2[i]] = aux[c2[j]];

```
movb -13(%ebp), %al
leal -40(%ebp), %ecx // 2*aux
addl $4, %ecx
addl -48(%ebp), %ecx // i+...
movb %al, (%ecx)
```

G) pushl %esi

... jmp for

for: popl %esi

H) movl -40(%ebp), %eax // aux

cmpl 16(%ebp), %eax // aux != y

je else

movl -48(%ebp), %ecx // i - aux

jmp end

else: movl -44(%ebp), %ecx // j - aux

end: movl %ecx, -4(%ebp)

movl 8(%ebp), %ecx

movl 8(%ebp), %ecx

r: cmpl 16(%ebp), %eax

jle for

cmpl 4000(%ebp), %eax

jle for

imull \$40, %eax, %edx

addl %edx, %edx

movl %edx, %esi

movl 36(%esi), %edx

addl %eax, %esi

movl 12(%esi), %edx

addl \$5, %edx

...