

1.) El código 3.5 imprime el mensaje: “Hola desde el proceso x de y” siendo “x” el número del proceso que ha escrito el mensaje e “y” el total que le hemos establecido en el comando de ejecución por terminal.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾ lun, 21 de may, 18:32 1
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI

Archivo Editar Ver Buscar Terminal Ayuda
Hola desde el proceso 3 de 4
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 4 ./c5
Hola desde el proceso 0 de 4
Hola desde el proceso 2 de 4
Hola desde el proceso 3 de 4
Hola desde el proceso 1 de 4
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 4 ./c5
Hola desde el proceso 1 de 4
Hola desde el proceso 3 de 4
Hola desde el proceso 0 de 4
Hola desde el proceso 2 de 4
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 8 ./c5
Hola desde el proceso 6 de 8
Hola desde el proceso 4 de 8
Hola desde el proceso 5 de 8
Hola desde el proceso 1 de 8
Hola desde el proceso 3 de 8
Hola desde el proceso 2 de 8
Hola desde el proceso 7 de 8
Hola desde el proceso 0 de 8
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 12 ./c5
Hola desde el proceso 3 de 12
Hola desde el proceso 7 de 12
Hola desde el proceso 10 de 12
Hola desde el proceso 11 de 12
Hola desde el proceso 0 de 12
Hola desde el proceso 8 de 12
Hola desde el proceso 2 de 12
Hola desde el proceso 9 de 12
Hola desde el proceso 5 de 12
Hola desde el proceso 6 de 12
Hola desde el proceso 4 de 12
Hola desde el proceso 1 de 12
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 2 ./c5
Hola desde el proceso 0 de 2
Hola desde el proceso 1 de 2
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI#
```

En el código 3.6, si no es el hilo con identificador 0, imprime “Proceso x de y\nSaludos desde el proceso x” siendo “x” el número del proceso actual e “y” el total de procesos. Al tener el método MPI_Send, este hilo se bloquea hasta que el otro reciba.

En caso de que el hilo tenga el identificador 0, imprime “Proceso 0, total de procesos = x”, siendo “x” el número total de hilos. Al tener el método MPI_Recv, este hilo se bloquea hasta que recibe algo.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾ lun, 21 de may, 18:44 1
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI

Archivo Editar Ver Buscar Terminal Ayuda
Proceso 1 de 12
Proceso 11 de 12
proceso 0, total procesos = 12 Saludos desde el proceso 1!
Proceso 2 de 12
Saludos desde el proceso 2!
Saludos desde el proceso 3!
Saludos desde el proceso 4!
Saludos desde el proceso 5!
Saludos desde el proceso 6!
Saludos desde el proceso 7!
Saludos desde el proceso 8!
Saludos desde el proceso 9!
Saludos desde el proceso 10!
Saludos desde el proceso 11!
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 4 ./c6
proceso 0, total procesos = 4 Proceso 2 de 4
Proceso 1 de 4
Proceso 3 de 4
Saludos desde el proceso 1!
Saludos desde el proceso 2!
Saludos desde el proceso 3!
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 8 ./c6
proceso 0, total procesos = 8 Proceso 1 de 8
Proceso 4 de 8
Proceso 2 de 8
Proceso 7 de 8
Saludos desde el proceso 1!
Saludos desde el proceso 2!
Proceso 5 de 8
Proceso 3 de 8
Saludos desde el proceso 3!
Saludos desde el proceso 4!
Saludos desde el proceso 5!
Proceso 6 de 8
Saludos desde el proceso 6!
Saludos desde el proceso 7!
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI#
```

2.) Para este ejercicio, declaramos la variable myrank como identificador de los procesos ingresados mediante el comando de ejecución del programa y, cuando el identificador sea 0 imprimirá “Hola amigos, soy vuestro maestro con id myrank”, y, si myrank es distinto de 0 imprimirá “Hola maestro, soy el esclavo myrank”.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾ lun, 21 de may, 19:01 1
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11

Archivo Editar Ver Buscar Terminal Ayuda
Hola amigos, soy vuestro maestro con id 0
Hola maestro, soy el esclavo 1
Hola maestro, soy el esclavo 3
Hola maestro, soy el esclavo 2
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11# mpirun -n 8 ./el
Hola maestro, soy el esclavo 1
Hola maestro, soy el esclavo 3
Hola maestro, soy el esclavo 4
Hola maestro, soy el esclavo 5
Hola maestro, soy el esclavo 7
Hola amigos, soy vuestro maestro con id 0
Hola maestro, soy el esclavo 2
Hola maestro, soy el esclavo 6
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11# mpirun -n 12 ./el
Hola maestro, soy el esclavo 2
Hola maestro, soy el esclavo 5
Hola maestro, soy el esclavo 7
Hola maestro, soy el esclavo 9
Hola maestro, soy el esclavo 1
Hola maestro, soy el esclavo 4
Hola maestro, soy el esclavo 6
Hola maestro, soy el esclavo 11
Hola amigos, soy vuestro maestro con id 0
Hola maestro, soy el esclavo 8
Hola maestro, soy el esclavo 3
Hola maestro, soy el esclavo 10
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11# mpirun -n 4 ./el
Hola amigos, soy vuestro maestro con id 0
Hola maestro, soy el esclavo 1
Hola maestro, soy el esclavo 2
Hola maestro, soy el esclavo 3
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11#
```

3.) En el código 3.7, en el cálculo de PI: Se crean dos variables “myid” y “numprocs” que contienen, respectivamente, el id del proceso actual y del número total de procesos. Luego, por cada proceso, un bucle se reitera 100 veces dejando el valor obtenido de la función “f()” en la variable sum (suma parcial). Cuando “myid” es 0, el proceso 0 recibe del resto de procesos sus valores de pi. Y cuando “myid” es distinto de 0, el resto de procesos envían al proceso 0 su valor de pi. Finalmente, es el proceso 0 el que imprime el resultado con el error cometido.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾ lun, 21 de may, 19:13 1
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI

Archivo Editar Ver Buscar Terminal Ayuda
Processor: 2, mypi: 0.7829244650957667
Processor: 0, mypi: 0.7928763409009609
pi es aproximadamente 3.1416009869231249, el error cometido es 0.0000083333333318
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 4 ./c7
Processor 1 of 4
Processor: 1, mypi: 0.7879260283629755
Processor 0 of 4
Processor 2 of 4
Processor: 2, mypi: 0.7829244650957667
Processor: 0, mypi: 0.7928763409009609
Processor 3 of 4
Processor: 3, mypi: 0.7778741525634219
pi es aproximadamente 3.1416009869231249, el error cometido es 0.0000083333333318
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 4 ./c7
Processor 3 of 4
Processor: 3, mypi: 0.7778741525634219
Processor 0 of 4
Processor: 0, mypi: 0.7928763409009609
Processor 1 of 4
Processor: 1, mypi: 0.7879260283629755
Processor 2 of 4
Processor: 2, mypi: 0.7829244650957667
pi es aproximadamente 3.1416009869231249, el error cometido es 0.0000083333333318
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI#
```

En el código 3.7, en el cálculo de un área: En la zona paralela, tenemos un bucle que calcula la base parcial por cada proceso del programa y, luego, se imprime dicha área. Luego, si la variable “mydi” (correspondiente con el identificador del proceso) es distinto que 0, el proceso envía al proceso 0 el valor de su área. Si “myid” es igual a 0, el proceso 0 recibe de los otros procesos el valor de sus áreas locales e imprime el valor del área total.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾ lun, 21 de may, 19:36 1
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI

Archivo Editar Ver Buscar Terminal Ayuda
El thread 1 calcula la iteracion i = 3
El thread 1 calcula la iteracion i = 5
El thread 1 calcula la iteracion i = 7
El thread 1 calcula la iteracion i = 9
Processor: 1, mi_area: 60
El thread 0 calcula la iteracion i = 0
El thread 0 calcula la iteracion i = 2
El thread 0 calcula la iteracion i = 4
El thread 0 calcula la iteracion i = 6
El thread 0 calcula la iteracion i = 8
Processor: 0, mi_area: 50
*****TOTAL: 110 *****
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 4 ./c7
El thread 0 calcula la iteracion i = 0
El thread 0 calcula la iteracion i = 4
El thread 0 calcula la iteracion i = 8
Processor: 0, mi_area: 30
El thread 1 calcula la iteracion i = 1
El thread 1 calcula la iteracion i = 5
El thread 1 calcula la iteracion i = 9
Processor: 1, mi_area: 36
El thread 3 calcula la iteracion i = 3
El thread 3 calcula la iteracion i = 7
Processor: 3, mi_area: 24
El thread 2 calcula la iteracion i = 2
El thread 2 calcula la iteracion i = 6
Processor: 2, mi_area: 20
*****TOTAL: 110 *****
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI#
```

4.) Para este ejercicio hacemos que, inicialmente, sea el proceso 0 sea el que ejecute el producto escalar de dos vectores. Luego, establecemos una barrera con “MPI_Barrier (MPI_COMM_WORLD);” y, a continuación, todos los procesos empezarán a la vez con el calculo paralelo del producto escalar.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾ jue, 24 de may, 10:10 1
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11

Archivo Editar Ver Buscar Terminal Ayuda
El thread 1, de 4 threads, calcula la iteracion i = 9: 9*9=81
Proceso: 1, mi_suma: 107
El thread 3, de 4 threads, calcula la iteracion i = 3: 3*3=9
El thread 3, de 4 threads, calcula la iteracion i = 7: 7*7=49
Proceso: 3, mi_suma: 58
La suma paralela es: 285
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11# mpirun -n 4 ./e2
Primero realizaremos la suma secuencial:
La suma secuencial es: 285
A continuacion, la suma paralela:
El thread 0, de 4 threads, calcula la iteracion i = 0: 0*0=0
El thread 0, de 4 threads, calcula la iteracion i = 4: 4*4=16
El thread 0, de 4 threads, calcula la iteracion i = 8: 8*8=64
Proceso: 0, mi_suma: 80
El thread 1, de 4 threads, calcula la iteracion i = 1: 1*1=1
El thread 2, de 4 threads, calcula la iteracion i = 2: 2*2=4
El thread 2, de 4 threads, calcula la iteracion i = 6: 6*6=36
Proceso: 2, mi_suma: 40
El thread 3, de 4 threads, calcula la iteracion i = 3: 3*3=9
El thread 3, de 4 threads, calcula la iteracion i = 7: 7*7=49
Proceso: 3, mi_suma: 58
El thread 1, de 4 threads, calcula la iteracion i = 5: 5*5=25
El thread 1, de 4 threads, calcula la iteracion i = 9: 9*9=81
Proceso: 1, mi_suma: 107
La suma paralela es: 285
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11#
```


5.) En el código 3.7, en el cálculo de PI: En este código, en vez de enviar los datos al proceso root (entendiendo como root el proceso 0), todos ejecutan la instrucción “MPI_Reduce(. .)” por lo que nos ahorramos el hecho de enviar los datos al root ya que de esto se encarga esta directiva.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI
Archivo Editar Ver Buscar Terminal Ayuda
Process 5 of 8
Process 1 of 88
Processo: 0, mypi: 0.4980544747081712
Processo: 2, mypi: 0.4555160142348754
Processo: 4, mypi: 0.3798219584569733
Processo: 6, mypi: 0.3011764705882353
Processo: 5, mypi: 0.3395225464190981
Processo: 1, mypi: 0.4830188679245283
Processo: 3, mypi: 0.4196721311475410
Processo: 7, mypi: 0.2661122661122661
pi es aproximadamente 3.1428947295916885, el error cometido es 0.0013020760018954
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 4 ./c8
Process 3 of 4
Process 0 of 4¿Cuantos intervalos quiere utilizar?
Process 2 of 4
Process 1 of 44
Processo: 0, mypi: 0.9846153846153847
pi es aproximadamente 3.1468005183939427, el error cometido es 0.0052078648041496
Processo: 1, mypi: 0.8767123287671232
Processo: 2, mypi: 0.7191011235955056
Processo: 3, mypi: 0.5663716814159292
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI#
```

En el código 3.7, en el cálculo de un área: En este código, y al igual que en el cálculo de PI, también se usa “MPI_Reduce(. .)” lo que nos ahorra el envío de los cálculos parciales al proceso root.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI
Archivo Editar Ver Buscar Terminal Ayuda
Processor: 2, mi_area: 20
El thread 3 calcula la iteracion i = 3
El thread 3 calcula la iteracion i = 7
Processor: 3, mi_area: 24
*****TOTAL: 110 *****
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI# mpirun -n 8 ./c8
El thread 0 calcula la iteracion i = 0
El thread 0 calcula la iteracion i = 8
Processor: 0, mi_area: 20
El thread 1 calcula la iteracion i = 1
El thread 1 calcula la iteracion i = 9
Processor: 1, mi_area: 24
El thread 2 calcula la iteracion i = 2
Processor: 2, mi_area: 6
El thread 3 calcula la iteracion i = 3
Processor: 3, mi_area: 8
El thread 4 calcula la iteracion i = 4
Processor: 4, mi_area: 10
El thread 6 calcula la iteracion i = 6
Processor: 6, mi_area: 14
El thread 5 calcula la iteracion i = 5
Processor: 5, mi_area: 12
El thread 7 calcula la iteracion i = 7
Processor: 7, mi_area: 16
*****TOTAL: 110 *****
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11/codigoMPI#
```

6.) En este ejercicio solo tenemos que modificar el “ejercicioMPI_3.c” para quitar el envío de los datos y dejar, solamente el “MPI_Bcast(. . .)” y el “MPI_Reduce(. . .)”. El funcionamiento del ejercicio es el mismo pero resulta más eficiente.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11
Archivo Editar Ver Buscar Terminal Ayuda
El thread 0, de 4 threads, calcula la iteracion i = 8: 8*8=64
Proceso: 0, mi_suma: 80
El thread 3, de 4 threads, calcula la iteracion i = 3: 3*3=9
El thread 3, de 4 threads, calcula la iteracion i = 7: 7*7=49
Proceso: 3, mi_suma: 58
La suma paralela es: 285
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11# mpirun -n 4 ./e3
Primero realizaremos la suma secuencial:
La suma secuencial es: 285
A continuacion, la suma paralela:
El thread 0, de 4 threads, calcula la iteracion i = 0: 0*0=0
El thread 0, de 4 threads, calcula la iteracion i = 4: 4*4=16
El thread 0, de 4 threads, calcula la iteracion i = 8: 8*8=64
Proceso: 0, mi_suma: 80
El thread 1, de 4 threads, calcula la iteracion i = 1: 1*1=1
El thread 1, de 4 threads, calcula la iteracion i = 5: 5*5=25
El thread 1, de 4 threads, calcula la iteracion i = 9: 9*9=81
Proceso: 1, mi_suma: 107
El thread 3, de 4 threads, calcula la iteracion i = 3: 3*3=9
El thread 2, de 4 threads, calcula la iteracion i = 2: 2*2=4
El thread 3, de 4 threads, calcula la iteracion i = 7: 7*7=49
Proceso: 3, mi_suma: 58
El thread 2, de 4 threads, calcula la iteracion i = 6: 6*6=36
Proceso: 2, mi_suma: 40
La suma paralela es: 285
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11#
```

7.) Para este ejercicio, en la parte paralela, dividimos la matriz en trozos de igual tamaño, para que cada proceso trabaje con su trozo sin interferir en el trozo de otro proceso. A la hora de ejecutarlo inicialmente, tuve problemas debido a que estaba pasándole la matriz inicial a todos los procesos, y no los trozos previamente comentados. Una vez zanjado el problema, el programa se ejecuta sin problemas.

```
Aplicaciones ▾ Lugares ▾ Terminal ▾
root@Kali: ~/Universidad/PPD/Prácticas/Práctica 11
Archivo Editar Ver Buscar Terminal Ayuda
El proceso 3, de 4 procesos, calcula la iteracion i = 0: 510+53*5=775
El proceso 3, de 4 procesos, calcula la iteracion i = 0: 775+54*6=1099
El proceso 3, de 4 procesos, calcula la iteracion i = 0: 1099+55*7=1484
El proceso 2, de 4 procesos, calcula la iteracion i = 0: 0+32*0=0
El proceso 3, de 4 procesos, calcula la iteracion i = 1: 0+56*0=0
El proceso 3, de 4 procesos, calcula la iteracion i = 1: 0+57*1=57
El proceso 3, de 4 procesos, calcula la iteracion i = 1: 57+58*2=173
El proceso 2, de 4 procesos, calcula la iteracion i = 0: 0+33*1=33
El proceso 2, de 4 procesos, calcula la iteracion i = 0: 33+34*2=101
El proceso 2, de 4 procesos, calcula la iteracion i = 0: 101+35*3=206
El proceso 2, de 4 procesos, calcula la iteracion i = 0: 206+36*4=350
El proceso 3, de 4 procesos, calcula la iteracion i = 1: 173+59*3=350
El proceso 3, de 4 procesos, calcula la iteracion i = 1: 350+60*4=590
El proceso 3, de 4 procesos, calcula la iteracion i = 1: 590+61*5=895
El proceso 3, de 4 procesos, calcula la iteracion i = 1: 895+62*6=1267
El proceso 2, de 4 procesos, calcula la iteracion i = 0: 350+37*5=535
El proceso 2, de 4 procesos, calcula la iteracion i = 0: 535+38*6=763
El proceso 2, de 4 procesos, calcula la iteracion i = 0: 763+39*7=1036
El proceso 3, de 4 procesos, calcula la iteracion i = 1: 1267+63*7=1708
El proceso 2, de 4 procesos, calcula la iteracion i = 1: 0+40*0=0
El proceso 2, de 4 procesos, calcula la iteracion i = 1: 0+41*1=41
El proceso 2, de 4 procesos, calcula la iteracion i = 1: 41+42*2=125
El proceso 2, de 4 procesos, calcula la iteracion i = 1: 125+43*3=254
El proceso 2, de 4 procesos, calcula la iteracion i = 1: 254+44*4=430
El proceso 2, de 4 procesos, calcula la iteracion i = 1: 430+45*5=655
El proceso 2, de 4 procesos, calcula la iteracion i = 1: 655+46*6=931
El proceso 2, de 4 procesos, calcula la iteracion i = 1: 931+47*7=1260
El proceso 1, de 4 procesos, calcula la iteracion i = 1: 415+30*6=595
El proceso 1, de 4 procesos, calcula la iteracion i = 1: 595+31*7=812
El resultado paralelo es:
140    364    588    812    1036    1260    1484    1708
root@Kali:~/Universidad/PPD/Prácticas/Práctica 11#
```

Como en la captura de pantalla no se ve bien el resultado, lo expongo a continuación:

Primero realizaemos la multiplicacion secuencial:

El resultado secuencial es:

140 364 588 812 1036 1260 1484 1708

A continuaci3n, realizaremos la multiplicacion paralela:

El proceso 0, de 4 procesos, calcula la iteracion $i = 0$: $0+0*0=0$

El proceso 1, de 4 procesos, calcula la iteracion $i = 0$: $0+16*0=0$

El proceso 1, de 4 procesos, calcula la iteracion $i = 0$: $0+17*1=17$

El proceso 0, de 4 procesos, calcula la iteracion $i = 0$: $0+1*1=1$

El proceso 0, de 4 procesos, calcula la iteracion $i = 0$: $1+2*2=5$

El proceso 1, de 4 procesos, calcula la iteracion $i = 0$: $17+18*2=53$

El proceso 1, de 4 procesos, calcula la iteracion $i = 0$: $53+19*3=110$

El proceso 0, de 4 procesos, calcula la iteracion $i = 0$: $5+3*3=14$

El proceso 0, de 4 procesos, calcula la iteracion $i = 0$: $14+4*4=30$

El proceso 1, de 4 procesos, calcula la iteracion $i = 0$: $110+20*4=190$

El proceso 1, de 4 procesos, calcula la iteracion $i = 0$: $190+21*5=295$

El proceso 0, de 4 procesos, calcula la iteracion $i = 0$: $30+5*5=55$

El proceso 0, de 4 procesos, calcula la iteracion $i = 0$: $55+6*6=91$

El proceso 1, de 4 procesos, calcula la iteracion $i = 0$: $295+22*6=427$

El proceso 0, de 4 procesos, calcula la iteracion $i = 0$: $91+7*7=140$

El proceso 1, de 4 procesos, calcula la iteracion $i = 0$: $427+23*7=588$

El proceso 0, de 4 procesos, calcula la iteracion $i = 1$: $0+8*0=0$

El proceso 0, de 4 procesos, calcula la iteracion $i = 1$: $0+9*1=9$

El proceso 0, de 4 procesos, calcula la iteracion $i = 1$: $9+10*2=29$

El proceso 1, de 4 procesos, calcula la iteracion $i = 1$: $0+24*0=0$

El proceso 0, de 4 procesos, calcula la iteracion $i = 1$: $29+11*3=62$

El proceso 0, de 4 procesos, calcula la iteracion $i = 1$: $62+12*4=110$

El proceso 1, de 4 procesos, calcula la iteracion $i = 1$: $0+25*1=25$

El proceso 1, de 4 procesos, calcula la iteracion $i = 1$: $25+26*2=77$

El proceso 0, de 4 procesos, calcula la iteracion $i = 1$: $110+13*5=175$

El proceso 0, de 4 procesos, calcula la iteracion $i = 1$: $175+14*6=259$

El proceso 1, de 4 procesos, calcula la iteracion $i = 1$: $77+27*3=158$

El proceso 0, de 4 procesos, calcula la iteracion $i = 1$: $259+15*7=364$

El proceso 1, de 4 procesos, calcula la iteracion $i = 1$: $158+28*4=270$

El proceso 1, de 4 procesos, calcula la iteracion $i = 1$: $270+29*5=415$

El proceso 3, de 4 procesos, calcula la iteracion $i = 0$: $0+48*0=0$

El proceso 3, de 4 procesos, calcula la iteracion $i = 0$: $0+49*1=49$

El proceso 3, de 4 procesos, calcula la iteracion $i = 0$: $49+50*2=149$

El proceso 3, de 4 procesos, calcula la iteracion $i = 0$: $149+51*3=302$

El proceso 3, de 4 procesos, calcula la iteracion $i = 0$: $302+52*4=510$

El proceso 3, de 4 procesos, calcula la iteracion $i = 0$: $510+53*5=775$

El proceso 3, de 4 procesos, calcula la iteracion $i = 0$: $775+54*6=1099$

El proceso 3, de 4 procesos, calcula la iteracion $i = 0$: $1099+55*7=1484$

El proceso 2, de 4 procesos, calcula la iteracion $i = 0$: $0+32*0=0$

El proceso 3, de 4 procesos, calcula la iteracion $i = 1$: $0+56*0=0$

El proceso 3, de 4 procesos, calcula la iteracion $i = 1$: $0+57*1=57$

El proceso 3, de 4 procesos, calcula la iteracion $i = 1$: $57+58*2=173$

El proceso 2, de 4 procesos, calcula la iteracion $i = 0$: $0+33*1=33$
El proceso 2, de 4 procesos, calcula la iteracion $i = 0$: $33+34*2=101$
El proceso 2, de 4 procesos, calcula la iteracion $i = 0$: $101+35*3=206$
El proceso 2, de 4 procesos, calcula la iteracion $i = 0$: $206+36*4=350$
El proceso 3, de 4 procesos, calcula la iteracion $i = 1$: $173+59*3=350$
El proceso 3, de 4 procesos, calcula la iteracion $i = 1$: $350+60*4=590$
El proceso 3, de 4 procesos, calcula la iteracion $i = 1$: $590+61*5=895$
El proceso 3, de 4 procesos, calcula la iteracion $i = 1$: $895+62*6=1267$
El proceso 2, de 4 procesos, calcula la iteracion $i = 0$: $350+37*5=535$
El proceso 2, de 4 procesos, calcula la iteracion $i = 0$: $535+38*6=763$
El proceso 2, de 4 procesos, calcula la iteracion $i = 0$: $763+39*7=1036$

El proceso 3, de 4 procesos, calcula la iteracion $i = 1$: $1267+63*7=1708$

El proceso 2, de 4 procesos, calcula la iteracion $i = 1$: $0+40*0=0$
El proceso 2, de 4 procesos, calcula la iteracion $i = 1$: $0+41*1=41$
El proceso 2, de 4 procesos, calcula la iteracion $i = 1$: $41+42*2=125$
El proceso 2, de 4 procesos, calcula la iteracion $i = 1$: $125+43*3=254$
El proceso 2, de 4 procesos, calcula la iteracion $i = 1$: $254+44*4=430$
El proceso 2, de 4 procesos, calcula la iteracion $i = 1$: $430+45*5=655$
El proceso 2, de 4 procesos, calcula la iteracion $i = 1$: $655+46*6=931$
El proceso 2, de 4 procesos, calcula la iteracion $i = 1$: $931+47*7=1260$

El proceso 1, de 4 procesos, calcula la iteracion $i = 1$: $415+30*6=595$

El proceso 1, de 4 procesos, calcula la iteracion $i = 1$: $595+31*7=812$

El resultado paralelo es:

140 364 588 812 1036 1260 1484 1708