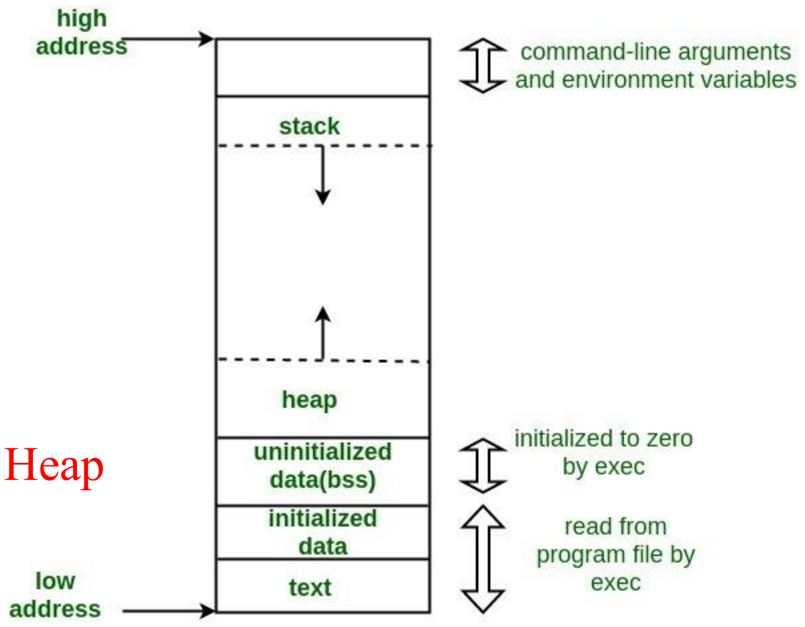
Exercise 2 Sol

Discussion CS140

Question 1. Memory allocated for the executable of a C program is divided into 4 areas: text, stack, heap, and data/BSS (for initialized/un-initialized global or static variables).

```
int  *r;   /* Line 1*/    r: BSS
void f(int *r){   /* Line 2*/  r: Stack
   int z=5;      /*  Line 3*/  z: Stack
   r[z]=4;       /*  Line 4*/  r[z]: Heap
}
void main(void){
   r=malloc(10*sizeof(int)); r: BSS, 10 * int: Heap
   f(r);  r: BSS
   free(r);  r: BSS
}
```



high address
command-line arguments and environment variables
stack
heap
uninitialized data(bss)
initialized to zero by exec
initialized data
read from program file by exec
text
low address

Mark the space location for variable r in Line 2, and array element r[z] in Line 4.

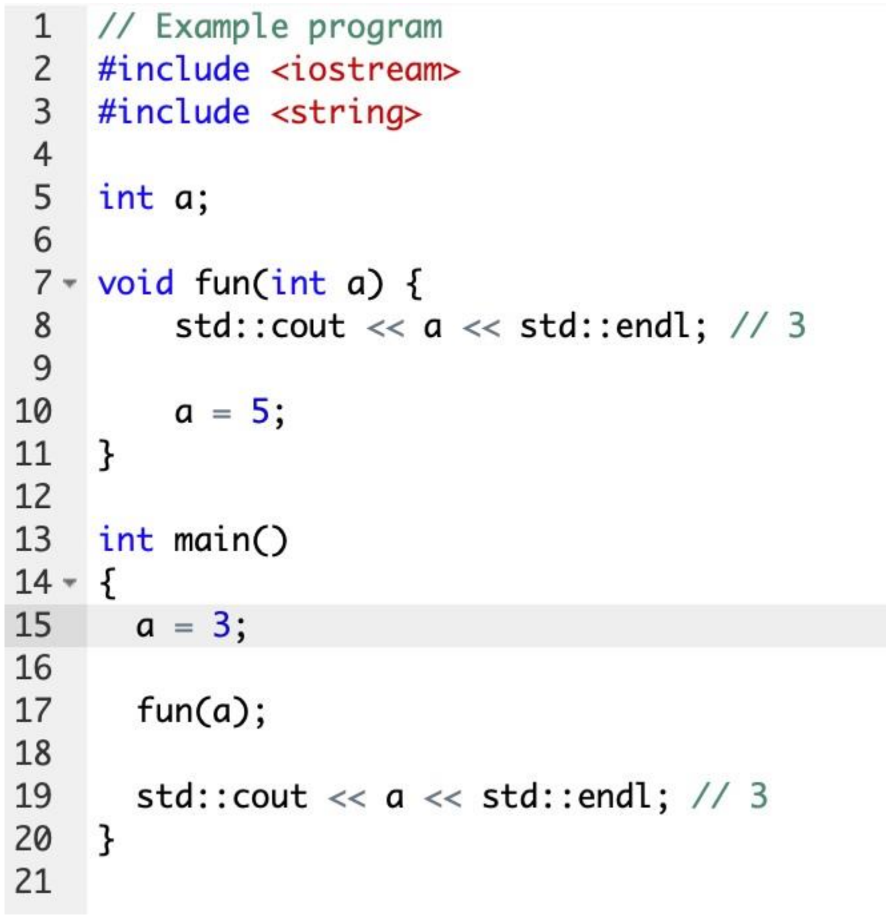
[✓ (a) stack, heap] (b) data/BSS, heap (c) stack, stack (d) data/BSS, stack (e) none of them is correct.

```
// Example program
#include <iostream>
#include <string>

int a;  BSS

int main()
{
  a = 3;  BSS

  std::cout << a << std::endl;
}
```

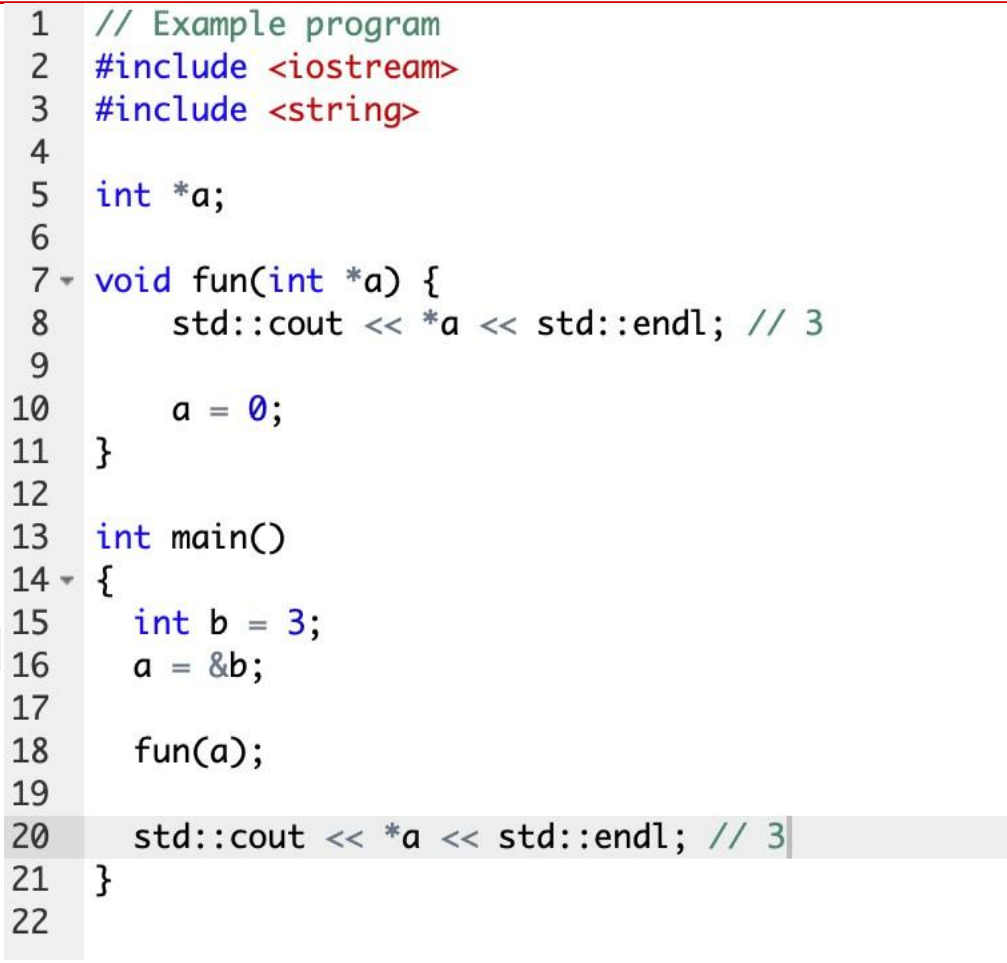
Output:
3

```
// Example program
#include <iostream>
#include <string>

int a;  BSS

int main()
{
  a = 3;  BSS

  {
      int a = 5;  Stack
      std::cout << a << std::endl;
  }

  std::cout << a << std::endl;
}
```

Output:
5
3

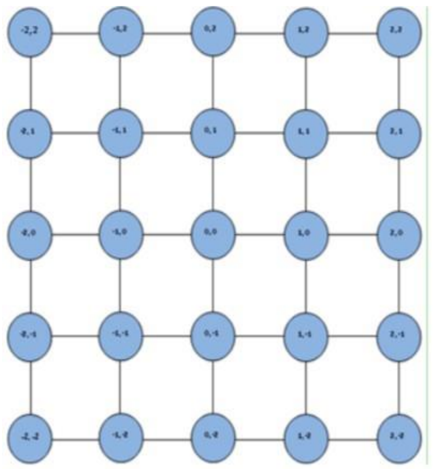


```cpp
// Example program
#include <iostream>
#include <string>

int a;

void fun(int a) {
    std::cout << a << std::endl; // 3

    a = 5;
}

int main()
{
  a = 3;

  fun(a);

  std::cout << a << std::endl; // 3
}

```



```cpp
// Example program
#include <iostream>
#include <string>

int *a;

void fun(int *a) {
    std::cout << *a << std::endl; // 3

    a = 0;
}

int main()
{
  int b = 3;
  a = &b;

  fun(a);

  std::cout << *a << std::endl; // 3
}

```

a[0] ? Stack



(a) 5Gb/s [✓ (b) 6Gb/s] (c) 7Gb/s (d) 8Gb/s (e) None of them is correct

Question 3. Assume the cache line (cache block) in a multiprocessor shared memory system is of size 64 bytes. Choose a statement from (a) to (d) below that is false or choose (e)

(a) When a processor executes a store instruction to write a word of 4 bytes to memory, this triggers the writing of a data block of 64 bytes from a local processor cache to memory.

(b) When a processor executes a load instruction of 4 bytes to read such a word from memory, this triggers the reading of a data block from memory to a local processor cache when there is a cache miss.

(c) Writing a data block to a processor cache has to invalidate copies of this block in other processor caches even these processors do not use this data block any more.

(d) False sharing can occur when multiple processors write data to different but consecutive data addresses in memory.

[✓ (e)] All of them are correct.

$$\begin{tikzcd}
x_{0,0} \arrow[r] \arrow[d] & x_{0,1} \arrow[r] \arrow[d] & x_{0,2} \arrow[r] \arrow[d] & x_{0,3} \arrow[d] \\
x_{1,0} \arrow[r] \arrow[d] & x_{1,1} \arrow[r] \arrow[d] & x_{1,2} \arrow[r] \arrow[d] & x_{1,3} \arrow[d] \\
x_{2,0} \arrow[r] & x_{2,1} \arrow[r] & x_{2,2} \arrow[r] & x_{2,3}
\end{tikzcd}$$

Q4: What is the degree of parallelism in this graph?

(a) 1 (b) 2 [✓ (c) 3] (b)4 (e) None of them is correct

Q5: What is the maximum speedup achievable in executing the above task graph in a parallel architecture?

(a) 1 [✓ (b) 2] (c) 3 (d) 4 (e) None of them is correct

Question 6.

(1) Assume a sequential program takes 1,000 seconds, what is the maximum speedup possible according to Amdahl's Law for a program that is 10% inherently serial and 90% fully parallelizable if you can only use 3 processors?

(2) Following the assumption of (1), if the parallelized program needs to spend an additional 100 seconds to exchange data and synchronize processors, what is the maximum speedup possible? Notice the additional 100 seconds are charged to every processor for data exchange and synchronization in this case.

Select one of the following 4 choices to answer (1) and (2):

(a) 3, 2 (b) 3, 2.5 [✓ (c) 2.5, 2] (d) 2.5, 2.5 (e) None of them is correct

(a) MPMD (Multiple program multiple data) is flexible as it allows different processing units run different code.

(b) In running SPMD (Single Program Multiple Data) code, each processing unit executes the same binary, but operates on different data based on processing unit ID.

(c) SPMD simplifies parallel programming because we write the same program for a different number of processing units.

[✓ (d)] At each time step of SPMD code execution, all processing units execute the exactly same machine instruction.

(e) All of them are true.

```
double n_over_p= (double) n/ ((double) noproc());
int me=mynode();
int first =
int last =
for( i=first;  i<=last && i<n; i++){
   compute(i);
}
```

Q8. Select a proper formula for variables first and last, controlling what is executed at each node.

(a) first = me*floor(n_over_p); last=(me+1)*floor(n_over_p)-1;
(b) first = me*floor(n_over_p)+1; last=(me+1)*floor(n_over_p);
[✓ (c)] first = me*ceil(n_over_p); last=(me+1)*ceil(n_over_p)-1;
(d) first = me*ceil(n_over_p)+1; last=(me+1)*ceil(n_over_p);

```
double n_over_p= (double) n/ ((double) noproc());
int me=mynode();
int first =
int last =
for( i=first;  i<=last && i<n; i++){
   compute(i);
}
```

Q9. Assume computation cost of the above code is dominated by function compute(i) with cost i+1 time units. Other cost such as loop overhead is considered as 0. When p=3, and n=6, what is the workload assigned to the first processor, second processor, and third processor? The workload of a processor is the sum of computation cost of all tasks assigned to this processor.

(a) 2, 2, 2
[✓ (b) 3, 7, 11]
(c) 7, 7, 7
(d) 5, 7, 9
(e) All of them are incorrect

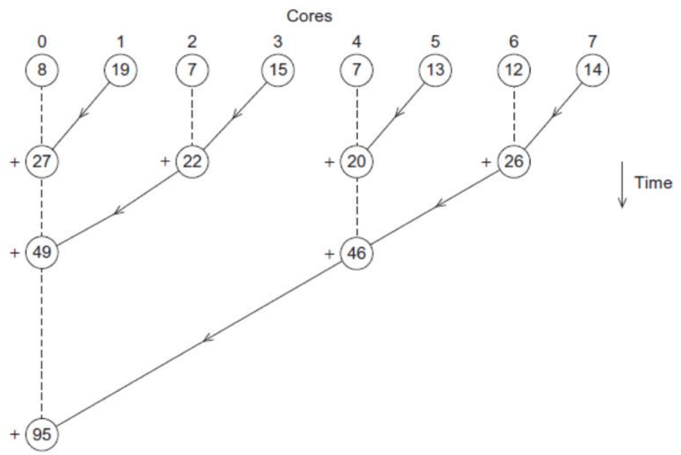
Q10: The following SPMD code uses a different mapping method (called cyclic mapping) to parallelize the sequential code.

```
int me=mynode();
p=noproc();
for(int  i=me;  i<n; i=i+p){
   compute(i);
}
```

Again assume computation cost of the above code is dominated by function compute(i) with cost i+1 time units. Other cost such as loop overhead is considered as 0. When p=3, and n=6, what is the workload assigned to the first processor, second processor, and third processor?

(a) 2, 2, 2
(b) 3, 7, 11
(c) 7, 7, 7
[✓ (d) 5, 7, 9]
(e) All of them are incorrect

```
me=mynode();
p= noproc();
sum=x[me];
if( (me %2 !=0  &&me>0)
          Send (sum, me -1);
For( gap=1; gap<p; gap =gap*2){
          if(me%(2*gap) ==0)){
                    Receive(other_sum, me+gap);
                    sum= other_sum+sum;
                    if( (me % (4*gap)) !=0  &&  me>=2*gap)
                              Send (sum, me -2*gap);
          }
}
```



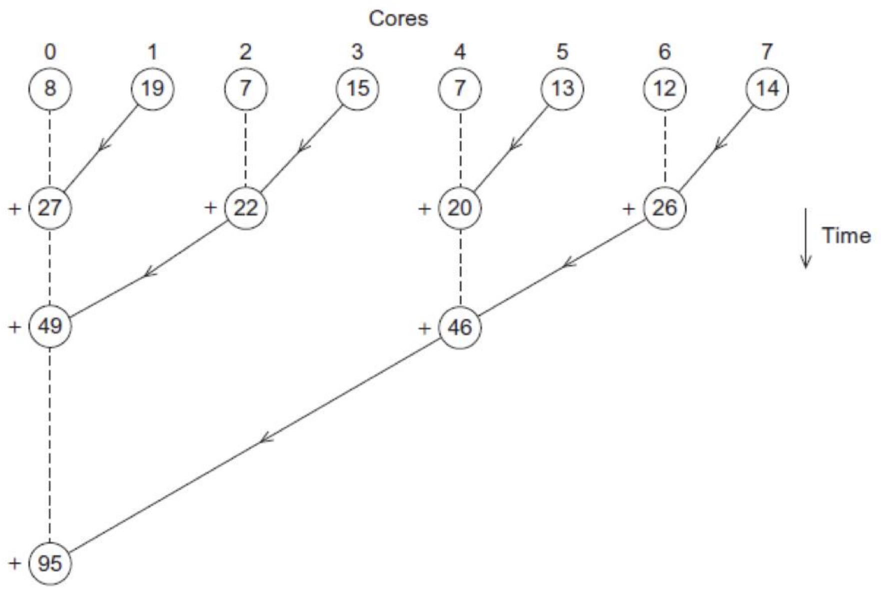
Q11: Select a true statement from the following four choices.

[✓ (a)] This code works correctly only when p is a power of 2

(b) This code works correctly only when p is not a power of 2

(c) This code works correctly for any integer p.

(d) None of above answers is correct.



Q12. What is the total number of receive functions called in the above SPMD code for processor 0 and processor 2 respectively? Assume p is a power of 2 and function log() uses base 2.

log(p), 1

Data Dependence for Parallelism Analysis

- Data dependence analysis
- Loop interchange

Discussion
UCSB CS140

Data Dependency

Given a program:

```
S1:    A=B+C;
S2:     B=C+A
```

Is there a flow dependency from S1 to S2?

Yes. A is defined in S1 and used in S2.

Read after Write

Is there an anti-dependence from S1 to S2?

Yes. B is used first, defined later

Write after Read

Is there output dependence from S1 to S2?

No. S1 &S2 write different memory locations.

Write after Write

Is there a flow or anti dependency from S2 to S1?

No. Otherwise the sequential program would be incorrect.

Data Dependency in a Loop Program

Given a Fibonacci program:

```
F[ 0 ]  =  F[ 1 ]  =  1;
for  (i  =  2;  i  <  n;  i++)
        F[ i ]  =  F[ i – 1 ] +F[ i – 2 ];
```

Call each iteration Si

Iteration space: S2, S3, …., Sn

Is there a flow dependence from S2 to S3?

Yes
S2: F[2]=F[1]+F[0]
S3: F[3]=F[2]+F[1]

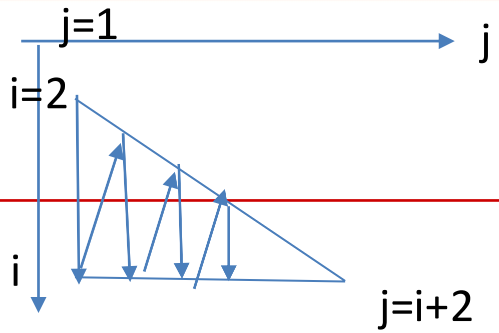
Is there an anti-dependence from S2 to S3?

No

Is there a flow or anti-dependence from S3 to S2?

No.
S2 runs before S3.

Loop Interchange Example



Given a sequential program:

```
for  i= 2 to n
    for  j = 1 to i+2
         F[ i ,j]  =  F[ i – 1,j ]
```

→

```
for  j= L1  to U1
    for  i = L2 to U2
         F[ i ,j]  =  F[ i – 1,j ]
```

All inequalities on i and j bounds: $\boxed{i \geq 2,\ i \leq n,\ j \geq 1,\ j \leq i+2}$

All inequalities on i bounds: $\boxed{j \geq 1,\ j \leq i+2 \rightarrow\ L1?\ U1?}$

$\boxed{\rightarrow\ L1=1.\ U1= n+2}$

All inequalities on i bounds: $\boxed{i \geq 2,\ i \leq n,\ j \leq i+2}$

$\boxed{\rightarrow i \geq 2, i \geq j-2.\ L2=\max(2, j-2)}$

$\boxed{\rightarrow\ i \leq n.\ U2=n}$