Bankers Algorithms in C++:

Code:

```
#include<iostream>
#include<iostream>
#include <iomanip>
using namespace std;
class Process{
  int total_resourses;
      int total_prcesses;
      int *max_resourses;
      int *allocated;
      int *available;
       bool *running_processes;
      int allocate_table[30][20];
      int max_claim_process[30][20];
       void get_max_resoures(){
             cout<<"Enter the Total MAX resoures Instances"<<endl;</pre>
             for(int i=0;i<total_resourses;i++){</pre>
                    cout<<"Enter R"<<i+1<<": ";cin>>max_resourses[i];
```

```
cout<<endl;
       }
       void print_max_resourses(){
              cout<<"Total MAX resoures Instances Are : "<<endl;</pre>
              for(int i=0;i<total_resourses;i++){</pre>
                      cout<<"R"<<i+1<<":
"<<setiosflags(ios::left)<<setw(10)<<max_resourses[i];
              }
              cout<<endl;
              line(70);
       }
       void line(int n){
              for(int a=1;a<=n;a++){
                     cout <<"_";
              cout<<endl;
       }
       void status_running_at_start(){
              for(int i=0 ;i<total_prcesses;i++){</pre>
                     running_processes[i] = true;
              }
       }
       void get_allocate_table(){
              cout<<"Enter the allocation table of the Processes !"<<endl;</pre>
```

```
for(int i=0;i<total_prcesses;i++){</pre>
               cout<<"Process "<<i+1<<endl;</pre>
               for(int j=0;j<total_resourses;j++){</pre>
                       cout<<"R"<<j+1<<" Allocated : ";cin>>allocate_table[i][j];
               }
               cout<<endl;
       }
}
void print_allocate_table(){
       cout<<''Allocated Table "<<endl;</pre>
       line(70);
       cout<<setiosflags(ios::left)<<setw(12)<<"Process";</pre>
       for(int p=1;p<=total_resourses;p++){</pre>
               cout<<"R"<<setiosflags(ios::left)<<setw(12)<<p;</pre>
       }
       cout<<endl;
       line(70);
       for(int i=0;i<total_prcesses;i++){</pre>
               cout<<setiosflags(ios::left)<<setw(12)<<i+1;</pre>
               for(int j=0;j<total_resourses;j++){</pre>
                       cout<<setiosflags(ios::left)<<setw(12)<<allocate_table[i][j];</pre>
               }
               cout<<endl;
```

```
line(70);
       }
       void get_max_claim_table(){
              cout<<"Enter the MAX Claim table of the Processes !"<<endl;</pre>
              for(int i=0;i<total_prcesses;i++){</pre>
                      cout<<"Process "<<i+1<<endl;
                      for(int j=0;j<total_resourses;j++){</pre>
                             cout<<"R"<<j+1<<" MAX Claim:
";cin>>max_claim_process[i][j];
                      }
                      cout<<endl;
              }
       }
       void print_max_claim_table(){
              cout<<"MAX Claim Table "<<endl;</pre>
              line(70);
              cout<<setiosflags(ios::left)<<setw(12)<<"Process";</pre>
              for(int p=1;p<=total_resourses;p++){</pre>
                      cout<<"R"<<setiosflags(ios::left)<<setw(12)<<p;</pre>
               }
              cout<<endl;
              line(70);
              for(int i=0;i<total_prcesses;i++){</pre>
                      cout<<setiosflags(ios::left)<<setw(12)<<i+1;</pre>
```

```
for(int j=0;j<total_resourses;j++){</pre>
       cout<<setiosflags(ios::left)<<setw(12)<<max_claim_process[i][j];</pre>
                       }
                       cout<<endl;
               }
               line(70);
       }
       void get_allocated_resourses(){
               for(int i=0;i<total_prcesses;i++){</pre>
                       for(int j=0;j<total_resourses;j++){</pre>
                              allocated[j] += allocate_table[i][j];
                       }
               }
       }
       void print_allocated_resoures(){
               cout<<''Allocated Resourse are : ''<<endl;</pre>
                       for(int i=0;i<total_resourses;i++){</pre>
                       cout<<"R"<<i+1<<":
"<<setiosflags(ios::left)<<setw(10)<<allocated[i];
               cout<<endl;
               line(70);
       }
       void get_available_resourses(){
```

```
for(int i=0;i<total_prcesses;i++){</pre>
                       for(int j=0; j< total\_resourses; j++)\{
                              available[j] = max_resourses[j] - allocated[j];
                       }
               }
       }
       void print_available_resoures(){
               cout<<''Available Resourse are : ''<<endl;</pre>
                       for(int i=0;i<total_resourses;i++){</pre>
                       cout<<"R"<<i+1<<":
"<<setiosflags(ios::left)<<setw(10)<<available[i];
               }
               cout<<endl;
               line(70);
       }
       void set_array(int *value,int size){
               for(int i=0;i<size;i++){
                       value[i]=0;
               }
       }
       void check_save(){
               int safe,exec,count=total_prcesses;
               while (count != 0){
      safe = 0;
      for (int i = 0; i < total_prcesses; i++){</pre>
```

```
if (running_processes[i]){
            exec = 1;
            for (int j = 0; j < total\_resourses; j++){
               if (max_claim_process[i][j] - allocate_table[i][j] > available[j]){
                  exec = 0;
                  break;
               }
            }
            if (exec){
                                             cout<<"Process "<< i + 1<<" is
executing"<<endl;
               running_processes[i] = false;
               count--;
               safe = 1;
               for (int j = 0; j < total\_resourses; j++){
                                                     available[j] += allocate_table[i][j];
               }
               break;
            }
         }
      }
     if (!safe) {
                        cout<<"The process is in unsafe state."<<endl;</pre>
```

```
break;
   }
   else {
      cout<<"Your process is in safe state"<<endl;</pre>
      cout<<"Available vector :";</pre>
      for (int i = 0; i < total_resourses; i++)</pre>
      {
         cout<< available[i] <<" ";</pre>
      }
      cout<<endl<<endl;
   }
}
    }
    public:
    Process(int resourses,int processes){
           total_prcesses = processes;
           total_resourses = resourses;
           max_resourses = new int [total_resourses];
           allocated = new int [total_resourses];
           available = new int [total_resourses];
           set_array(allocated,total_resourses);
           set_array(available,total_resourses);
```

```
get_max_resoures();
             running_processes = new bool[total_prcesses];
             status_running_at_start();
             get_allocate_table();
             get_max_claim_table();
             get_allocated_resourses();
             get_available_resourses();
             print_allocate_table();
             print_max_claim_table();
             print_max_resourses();
             print_allocated_resoures();
             print_available_resoures();
             check_save();
       }
};
int main(){
      int processes, resourses;
      cout<<"Enter Number of Processes: ";cin>>processes;
      cout<<"Enter Number of Resourses : ";cin>>resourses;
      Process p = Process(resourses,processes);
       return 0;
```

Output Example 1:

```
C:\Users\NAEEM UR RAHMAN\OneDrive\Desktop\Bankers Algorithm.exe
Enter Number of Processes : 4
Enter Number of Resourses : 2
Enter the Total MAX resoures Instances
Enter R1 : 12
Enter R2 : 12
Enter the allocation table of the Processes !
Process 1
R1 Allocated : 1
R2 Allocated : 1
Process 2
R1 Allocated : 3
R2 Allocated : 3
Process 3
R1 Allocated : 1
R2 Allocated : 2
Process 4
R1 Allocated : 1
R2 Allocated : 1
Enter the MAX Claim table of the Processes !
Process 1
R1 MAX Claim : 3
R2 MAX Claim : 3
Process 2
R1 MAX Claim : 4
R2 MAX Claim : 5
Process 3
R1 MAX Claim : 3
R2 MAX Claim : 3
Process 4
R1 MAX Claim : 3
R2 MAX Claim : 2
```

■ C:\U	sers\NAEEN	I UR RAHMAN\	OneDrive\Desktop\Bankers Algorithm.exe
Allocate	d Table		
Process	R1	R2	
1	1	1	
2	3	3	
3	1	2	
4	1	1	
MAX Clai	m Table		
Process	R1	R2	
1	3	3	
2	4	5	
3	3	3	
4	3	2	
Total MA	X resoures	Instances Are	: :
R1 : 12	R2	: 12	
Allocate R1 : 6	d Resourse R2		
Availabl	e Resourse	are :	
R1 : 6	R2		
	1 is execu		
Your pro	cess is in	safe state	
	e vector :		
	2 is execut		
		safe state	
	e vector :		
	3 is execut		
	cess is in e vector ::	safe state	
	e vector :: 4 is execu		
		safe state	
	e vector ::		
		, ,	

Output Example 2:

```
■ C:\Users\NAEEM UR RAHMAN\OneDrive\Desktop\Bankers Algorithm.exe
Enter Number of Processes : 5
Enter Number of Resourses : 3
Enter the Total MAX resoures Instances
Enter R1 : 5
Enter R2 : 5
Enter R3 : 5
Enter the allocation table of the Processes !
Process 1
R1 Allocated : 1
R2 Allocated : 0
R3 Allocated : 0
Process 2
R1 Allocated : 2
R2 Allocated : 2
R3 Allocated : 0
Process 3
R1 Allocated : 0
R2 Allocated : 0
R3 Allocated : 2
Process 4
R1 Allocated : 0
R2 Allocated : 0
R3 Allocated : 0
Process 5
R1 Allocated : 1
R2 Allocated : 1
R3 Allocated : 1
Enter the MAX Claim table of the Processes !
Process 1
R1 MAX Claim : 3
R2 MAX Claim : 3
R3 MAX Claim : 2
Process 2
R1 MAX Claim : 3
R2 MAX Claim : 3
R3 MAX Claim : 3
Process 3
R1 MAX Claim : 1
R2 MAX Claim : 1
R3 MAX Claim : 2
Process 4
R1 MAX Claim : 1
R2 MAX Claim : 1
R3 MAX Claim : 1
Process 5
R1 MAX Claim : 2
R2 MAX Claim : 2
R3 MAX Claim : 1
```

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Allocated	Table				
2	Process	R1	R2	R3		
8	1	1	9	0		
AX Claim Table Tocess R1 R2 R3 3 3 2 3 3 3 3 1 1 2 1 1 2 1 1 1 2 2 1 1 Total MAX resoures Instances Are: 1:5 R2:5 R3:5 Tlocated Resourse are: 1:4 R2:3 R3:3 Valiable Resourse are: 1:1 R2:2 R3:2 Trocess 3 is executing Our process is in safe state vailable vector:1 2 4 Trocess 2 is executing Our process is in safe state vailable vector:3 4 4 Trocess 4 is executing Our process is in safe state vailable vector:4 4 4 Trocess 4 is executing Our process is in safe state vailable vector:4 4 4 Trocess 5 is executing Our process is in safe state vailable vector:5 5 5						
AX Claim Table rocess R1 R2 R3 3 3 2 3 3 3 2 1 1 2 2 1 1 1 2 1 2 2 1 otal MAX resoures Instances Are: 1:5 R2:5 R3:5 llocated Resourse are: 1:4 R2:3 R3:3 vailable Resourse are: 1:1 R2:2 R3:2 rocess 3 is executing our process is in safe state vailable vector:1 2 4 rocess 2 is executing our process is in safe state vailable vector:3 4 4 rocess 1 is executing our process is in safe state vailable vector:4 4 4 rocess 4 is executing our process is in safe state vailable vector:4 4 4 rocess 5 is executing our process is in safe state vailable vector:5 5 5			0	2		
AX Claim Table Tocess R1 R2 R3 3 3 3 3 1 1 1 2 1 1 2 2 1 1 1 2 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 1 2 2 1 2 1 2		0	0	0		
a 3 3 2 2 3 3 3 3 3 1 1 1 1 2 2 1 1 1 1 1		1	1	1		
3 3 3 3 3 3 3 1 1 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1	AX Claim	Table				
3 3 3 3 3 3 3 1 1 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1	Incore	01	p.,			
3 3 3 3 1 1 1 2 1 1 1 1 2 1 1 1 1 2 2 2 2	rocess	К1		кэ		
1 1 1 1 2 1 1 1 1 2 2 1 1 cotal MAX resoures Instances Are: 1:5 R2:5 R3:5						
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1						
otal MAX resoures Instances Are: 1:5 R2:5 R3:5 Illocated Resourse are: 1:4 R2:3 R3:3 vailable Resourse are: 1:1 R2:2 R3:2 rocess 3 is executing our process is in safe state vailable vector:1 2 4 rocess 2 is executing our process is in safe state vailable vector:3 4 4 rocess 1 is executing our process is in safe state vailable vector:4 4 4 rocess 4 is executing our process is in safe state vailable vector:4 4 4 rocess 5 is executing our process is in safe state vailable vector:5 5 5						
total MAX resoures Instances Are: 1:5 R2:5 R3:5						
1:5 R2:5 R3:5 llocated Resourse are: 1:4 R2:3 R3:3 vailable Resourse are: 1:1 R2:2 R3:2 rocess 3 is executing our process is in safe state vailable vector:1 2 4 rocess 2 is executing our process is in safe state vailable vector:3 4 4 rocess 1 is executing our process is in safe state vailable vector:4 4 4 rocess 4 is executing our process is in safe state vailable vector:4 4 4 rocess 5 is executing our process is in safe state vailable vector:5 5 5	•	2	2	1		
llocated Resourse are: 1:4 R2:3 R3:3 vailable Resourse are: 1:1 R2:2 R3:2 rocess 3 is executing our process is in safe state vailable vector:1 2 4 rocess 2 is executing our process is in safe state vailable vector:3 4 4 rocess 1 is executing our process is in safe state vailable vector:4 4 4 rocess 4 is executing our process is in safe state vailable vector:4 4 4 rocess 5 is executing our process is in safe state vailable vector:5 5 5						
1:4 R2:3 R3:3 vailable Resourse are: 1:1 R2:2 R3:2 rocess 3 is executing our process is in safe state vailable vector:1 2 4 rocess 2 is executing our process is in safe state vailable vector:3 4 4 rocess 1 is executing our process is in safe state vailable vector:4 4 4 rocess 4 is executing our process is in safe state vailable vector:4 4 4 rocess 5 is executing our process is in safe state vailable vector:5 5 5	R1 : 5	R2	: 5 R3 :	5		
vailable Resourse are: 1:1 R2:2 R3:2 rocess 3 is executing our process is in safe state vailable vector:1 2 4 rocess 2 is executing our process is in safe state vailable vector:3 4 4 rocess 1 is executing our process is in safe state vailable vector:4 4 4 rocess 4 is executing our process is in safe state vailable vector:4 4 4 rocess 5 is executing our process is in safe state vailable vector:5 5 5	llocated	Resourse	e are :			
1 : 1 R2 : 2 R3 : 2 rocess 3 is executing our process is in safe state vailable vector :1 2 4 rocess 2 is executing our process is in safe state vailable vector :3 4 4 rocess 1 is executing our process is in safe state vailable vector :4 4 4 rocess 4 is executing our process is in safe state vailable vector :4 4 4 rocess 5 is executing our process is in safe state vailable vector :4 5 5 rocess 5 is executing our process is in safe state vailable vector :5 5 5	1 : 4	R2	: 3 R3 :	3		
1 : 1 R2 : 2 R3 : 2 rocess 3 is executing our process is in safe state vailable vector :1 2 4 rocess 2 is executing our process is in safe state vailable vector :3 4 4 rocess 1 is executing our process is in safe state vailable vector :4 4 4 rocess 4 is executing our process is in safe state vailable vector :4 4 4 rocess 5 is executing our process is in safe state vailable vector :4 5 5 rocess 5 is executing our process is in safe state vailable vector :5 5 5	Available	Resourse				
our process is in safe state vailable vector :1 2 4 rocess 2 is executing our process is in safe state vailable vector :3 4 4 rocess 1 is executing our process is in safe state vailable vector :4 4 4 rocess 4 is executing our process is in safe state vailable vector :4 4 4 rocess 5 is executing our process is in safe state vailable vector :4 5 5	R1 : 1			2		
our process is in safe state vailable vector :1 2 4 rocess 2 is executing our process is in safe state vailable vector :3 4 4 rocess 1 is executing our process is in safe state vailable vector :4 4 4 rocess 4 is executing our process is in safe state vailable vector :4 4 4 rocess 5 is executing our process is in safe state vailable vector :4 5 5	Process 3	is execu	uting			
rocess 2 is executing our process is in safe state vailable vector :3						
our process is in safe state vailable vector :3	Available	vector :	1 2 4			
our process is in safe state vailable vector :3	Process 2	is execu	iting			
vailable vector :3						
rocess 1 is executing our process is in safe state vailable vector :4						
our process is in safe state vailable vector :4						
vailable vector :4 4 4 rocess 4 is executing our process is in safe state vailable vector :4 4 4 rocess 5 is executing our process is in safe state vailable vector :5 5 5						
rocess 4 is executing our process is in safe state vailable vector :4						
our process is in safe state vailable vector :4	Available	vector :	4 4 4			
our process is in safe state vailable vector :4	Dancare 1	i	tion			
vailable vector :4 4 4 rocess 5 is executing our process is in safe state vailable vector :5 5 5 rocess exited after 226.7 seconds with return value 0						
rocess 5 is executing our process is in safe state vailable vector :5 5 5 rocess exited after 226.7 seconds with return value 0						
our process is in safe state vailable vector :5	Avallable	AECEOL :	+ + +			
our process is in safe state vailable vector :5	Process 5	is execu	uting			
vailable vector :5 5 5 rocess exited after 226.7 seconds with return value 0						
	Process ev	ited aft	er 226.7 seconds	with ret	urn value 0	
		_,				