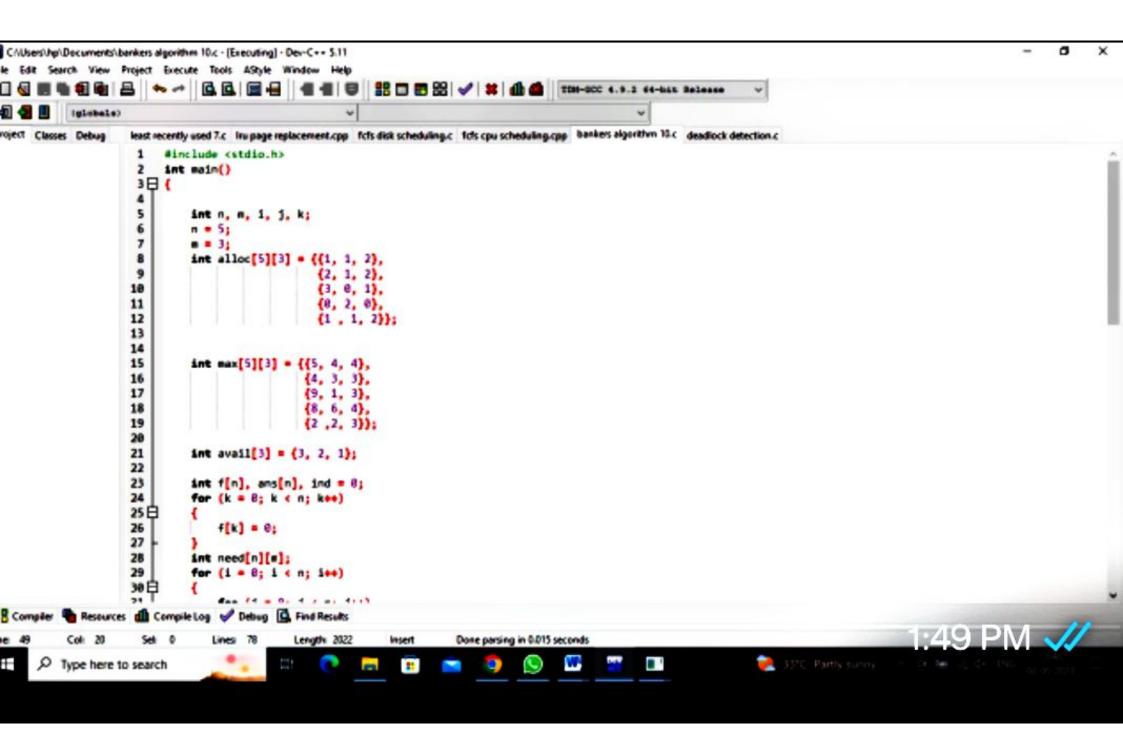
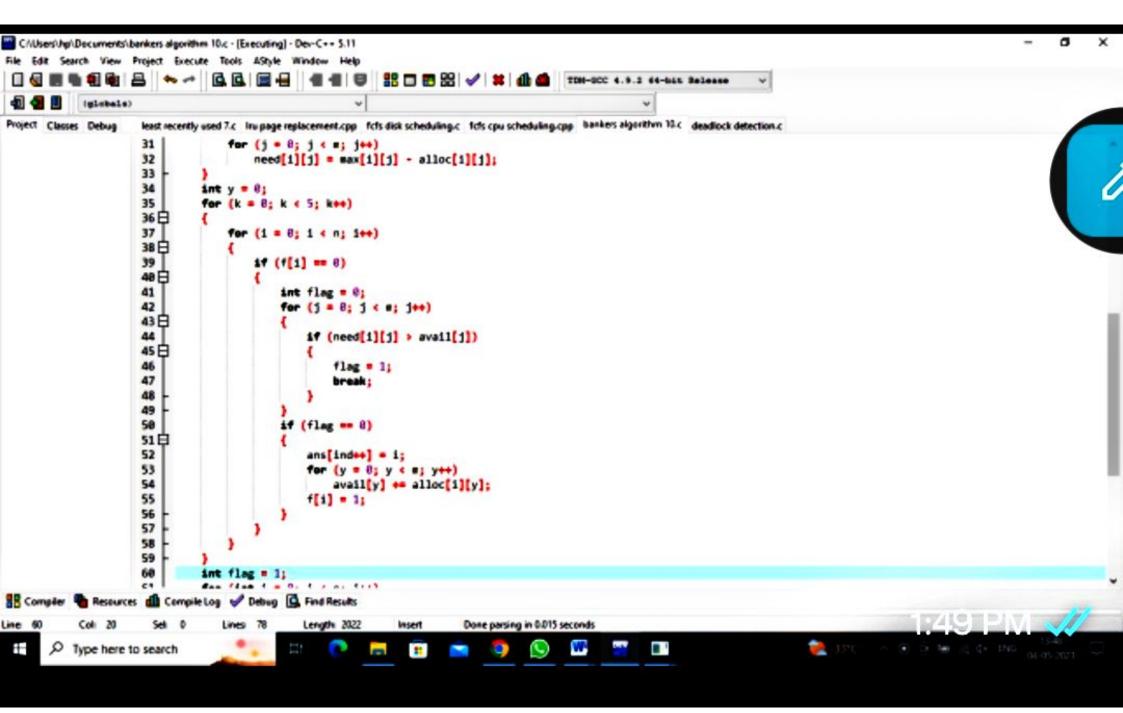
10. Consider the following process table with number of processes that contains allocation field (for showing the number of resources of type: A, B and C allocated to each process in the table), max field (for showing the maximum number of resources of type: A, B, and C that can be allocated to each process). Write a program to calculate the entries of need matrix using the formula: (Need)i = (Max)i - (Allocation)i

Process	Allocation	Max	Availble
	A B C	A B C	A B C
PO	1 1 2	5 4 4	3 2 1
P1	2 1 2	4 3 3	
P2	3 0 1	9 1 3	

P3	0 2 0	8 6 4
P4	1 12	2 2 3

3





```
C:\Users\np\Documents\bankers algorithm IU.C - [Executing] - Dev-C++ 5.11
le Edit Search View Project Execute Tools AStyle Window Help
                                                                                         TDM-GCC 4.9.2 64-bit Release
           (globals)
                    least recently used 7.c Iru page replacement.cpp fcfs disk scheduling.c fcfs cpu scheduling.cpp bankers algorithm 10.c deadlock detection.c
roject Classes Debug
                    49
                    50
                                           if (flag == 0)
                    51 🗎
                    52
                                               ans[ind++] = i:
                    53
                                               for (y = 0; y < m; y++)
                    54
                                                   avail[y] += alloc[i][y];
                    55
                                               f[i] = 1;
                    56
                    57
                    58
                    59
                    60
                              int flag = 1;
                    61
                              for (int i = 0; i < n; i++)
                    62 🖨
                    63
                                  if (f[i] == 0)
                    64 🖹
                    65
                                      flag = 0;
                    66
                                      printf("The following system is not safe");
                    67
                                      break:
                    68
                    69
                    70
                              if (flag == 1)
                    71 🖨
                                  printf("Following is the SAFE Sequence\n");
                    72
                    73
                                  for (i = 0; i < n - 1; i++)
                                      printf(" P%d ->", ans[i]);
                    74
                    75
                                  printf(" P%d", ans[n - 1]);
                    76
                    77
                              return (0);
Compiler 🖷 Resources 🛍 Compile Log 🥒 Debug 🚨 Find Results
          Col: 4
                                              Length: 2022
                                                                         Done parsing in 0.015 seconds
                                 Lines: 78
                                                              Insert
                                                                                                                         W
                                                                                                       Type here to search
```

C: Users hip Documents bankers algorithm Tuleve (3)

Following is the SAFE Sequence P1 -> P4 -> P0 -> P2 -> P3

Process exited after 0.03684 seconds with return value 0 Press any key to continue . . .





















































