Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology, Pune-37

(Anautonomous Institute of Savitribai Phule Pune University)

Department of Computer Engineering

Division	**CS**
Batch	**B1**
Roll no.	**90**
Name	**Aditya Shrinivas Kurapati**
PRN No	**12320184**

Bankers Algorithm Implementation Using C Program :

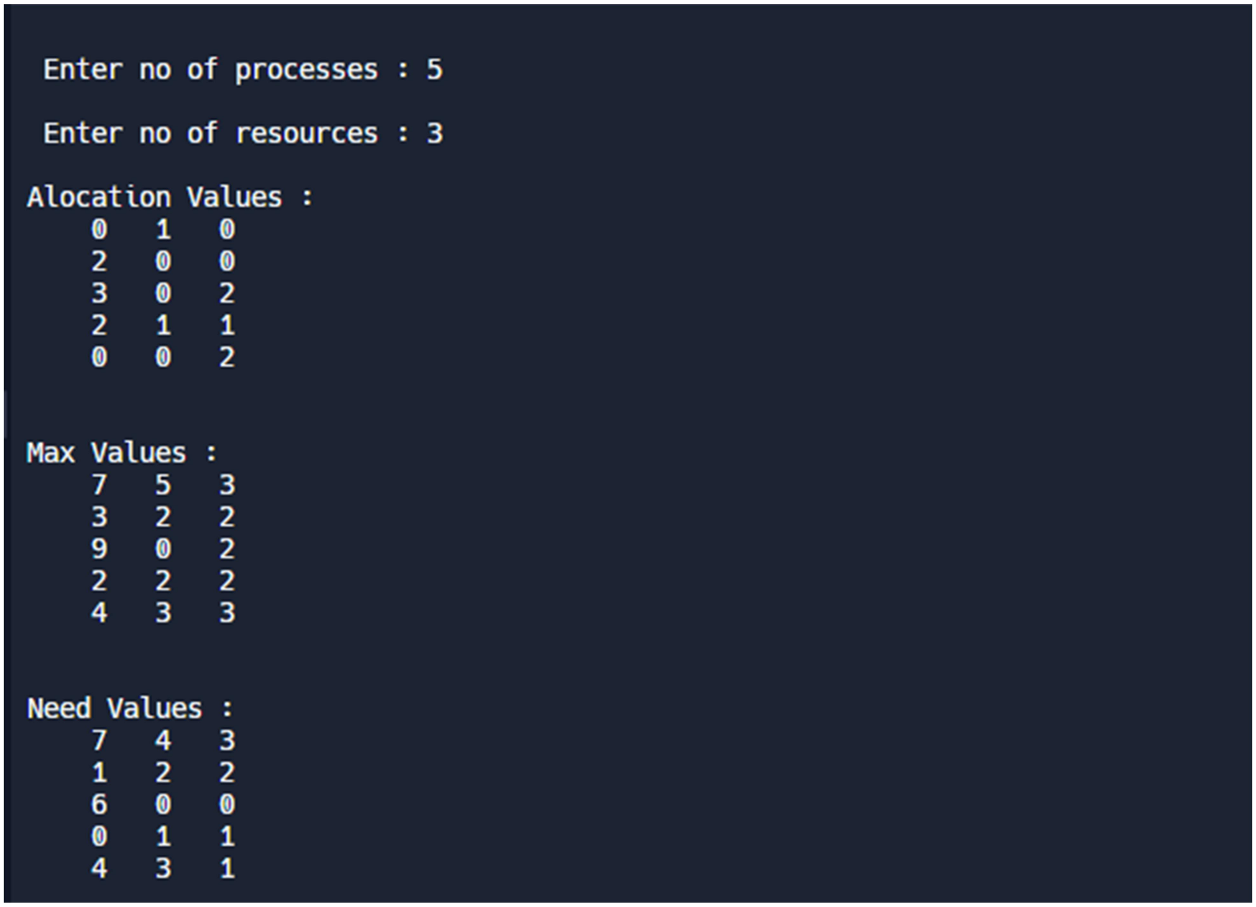
```c
#include <stdio.h>


int m, n, i, j, need[10][10], temp, z, y, p, k;

int al[5][3] = {{0, 1, 0}, {2, 0, 0}, {3, 0, 2}, {2, 1, 1}, {0, 0, 2}};

int max[10][10] = {{7, 5, 3}, {3, 2, 2}, {9, 0, 2}, {2, 2, 2}, {4, 3, 3}};

int av[10] = {3, 3, 2};


void main() {


  printf("\n Enter no of processes : ");

  scanf("%d", &m);


  printf("\n Enter no of resources : ");

  scanf("%d", &n);


  // Calculate the need matrix

  for (i = 0; i < m; i++) {

    for (j = 0; j < n; j++) {

      need[i][j] = max[i][j] - al[i][j];

    }

  }
```

```
// Print allocation values
printf("\nAlocation Values :\n");
for (i = 0; i < m; i++) {
  for (j = 0; j < n; j++) {
    printf("\t%d", al[i][j]);
  }
  printf("\n");
}

printf("\n\n");

// Print max values
printf("Max Values :\n");
for (i = 0; i < m; i++) {
  for (j = 0; j < n; j++) {
    printf("\t%d", max[i][j]);
  }
  printf("\n");
}

printf("\n\n");

// Print need values
```

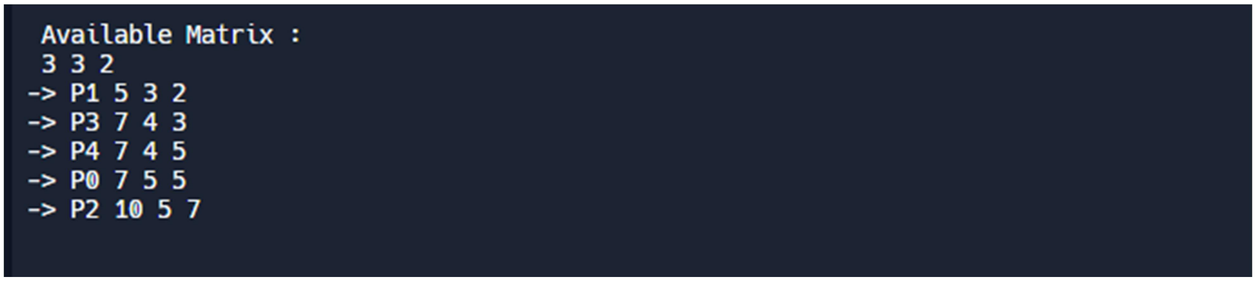
```
printf("Need Values :\n");
for (i = 0; i < m; i++) {
  for (j = 0; j < n; j++) {
    printf("\t%d", need[i][j]);
  }
  printf("\n");
}

printf("\n Available Matrix : \n ");

for (k = 0; k < n; k++) {
  printf("%d ", av[k]);
}
printf("\n");
p = 1;
y = 0;

while (p != 0) {
  p = 0; // Resetting p

  for (i = 0; i < m; i++) {
    z = 0;
    for (j = 0; j < n; j++) {
      if (need[i][j] <= av[j]) {
        z++;
      }
    }

    if (z == n && need[i][0] != -1) {
      printf("-> P%d ", i);
      y++;
      need[i][0] = -1;

      for (k = 0; k < n; k++) {
        av[k] += al[i][k];
        printf("%d ", av[k]);
      }
      printf("\n");
      p = 1; // Set p to 1 to indicate progress
    }
  }
}

if (y != m) {
  printf("\nSystem is in unsafe state\n");
}
```

```
  printf("\n");
}
```

OUTPUT:-



```
 Enter no of processes : 5

 Enter no of resources : 3

Alocation Values :
    0   1   0
    2   0   0
    3   0   2
    2   1   1
    0   0   2


Max Values :
    7   5   3
    3   2   2
    9   0   2
    2   2   2
    4   3   3


Need Values :
    7   4   3
    1   2   2
    6   0   0
    0   1   1
    4   3   1
```



```
 Available Matrix :
 3 3 2
-> P1 5 3 2
-> P3 7 4 3
-> P4 7 4 5
-> P0 7 5 5
-> P2 10 5 7
```