

OS LAB 09

Question 1: Implement the above code and paste the screen shot of the output.

Solution:

```
#include <stdio.h>

int main() {
    int p[10], np, b[10], nb, ch;
    int c[10], d[10], alloc[10], flag[10];
    int i, j;

    printf("\nEnter the number of processes: ");
    scanf("%d", &np);

    printf("Enter the number of memory blocks: ");
    scanf("%d", &nb);

    printf("\nEnter the size of each process:\n");
    for (i = 0; i < np; i++) {
        printf("Process %d: ", i);
        scanf("%d", &p[i]);
    }

    printf("\nEnter the size of each block:\n");
    for (j = 0; j < nb; j++) {
        printf("Block %d: ", j);
        scanf("%d", &b[j]);
        c[j] = b[j];
        d[j] = b[j];
    }

    if (np <= nb) {
        printf("\n1. First Fit\n2. Best Fit\n3. Worst Fit");

        do {
            printf("\n\nEnter your choice: ");
            scanf("%d", &ch);

            for (i = 0; i < np; i++) flag[i] = 1;
            for (j = 0; j < nb; j++) {
                b[j] = c[j];
                c[j] = d[j];
                d[j] = b[j];
            }
        }
    }
}
```

```
switch (ch) {
    case 1:
        printf("\n--- First Fit ---\n");
        for (i = 0; i < np; i++) {
            for (j = 0; j < nb; j++) {
                if (p[i] <= b[j]) {
                    alloc[i] = j;
                    printf("\nProcess %d of size %d is allocated in block %d of size %d", i, p[i], j, b[j]);
                    b[j] = 0;
                    flag[i] = 0;
                    break;
                }
            }
        }
        break;

    case 2:
        printf("\n--- Best Fit ---\n");

        for (i = 0; i < nb - 1; i++) {
            for (j = i + 1; j < nb; j++) {
                if (c[i] > c[j]) {
                    int temp = c[i];
                    c[i] = c[j];
                    c[j] = temp;
                }
            }
        }

        printf("After sorting block sizes (Best Fit):\n");
        for (i = 0; i < nb; i++) {
            printf("Block %d: %d\n", i, c[i]);
        }

        for (i = 0; i < np; i++) {
            for (j = 0; j < nb; j++) {
                if (p[i] <= c[j]) {
                    alloc[i] = j;
                    printf("\nProcess %d of size %d is allocated in block %d of size %d", i, p[i], j, c[j]);
                    c[j] = 0;
                    flag[i] = 0;
                    break;
                }
            }
        }
}
```

```
        break;

    case 3:
        printf("\n--- Worst Fit ---\n");

        for (i = 0; i < nb - 1; i++) {
            for (j = i + 1; j < nb; j++) {
                if (d[i] < d[j]) {
                    int temp = d[i];
                    d[i] = d[j];
                    d[j] = temp;
                }
            }
        }

        printf("After sorting block sizes (Worst Fit):\n");
        for (i = 0; i < nb; i++) {
            printf("Block %d: %d\n", i, d[i]);
        }

        for (i = 0; i < np; i++) {
            for (j = 0; j < nb; j++) {
                if (p[i] <= d[j]) {
                    alloc[i] = j;
                    printf("\nProcess %d of size %d is allocated in block  
%d of size %d", i, p[i], j, d[j]);
                    d[j] = 0;
                    flag[i] = 0;
                    break;
                }
            }
        }
        break;

    default:
        printf("Invalid choice!");
}

for (i = 0; i < np; i++) {
    if (flag[i]) {
        printf("\nProcess %d of size %d is not allocated.", i, p[i]);
    }
}

} while (ch <= 3);

} else {
    printf("Number of processes should not exceed number of blocks.\n");
```

Shayan
DT-22037

```
}  
  
    return 0;  
}
```

```
Enter the number of processes: 4  
Enter the number of memory blocks: 5
```

```
Enter the size of each process:  
Process 0: 212  
Process 1: 417  
Process 2: 112  
Process 3: 426
```

```
Enter the size of each block:  
Block 0: 100  
Block 1: 500  
Block 2: 200  
Block 3: 300  
Block 4: 600
```

```
1. First Fit  
2. Best Fit  
3. Worst Fit
```

```
Enter your choice: 1

--- First Fit ---

Process 0 of size 212 is allocated in block 1 of size 500
Process 1 of size 417 is allocated in block 4 of size 600
Process 2 of size 112 is allocated in block 2 of size 200
Process 3 of size 426 is not allocated.

Enter your choice: 2

--- Best Fit ---
After sorting block sizes (Best Fit):
Block 0: 100
Block 1: 200
Block 2: 300
Block 3: 500
Block 4: 600

Process 0 of size 212 is allocated in block 2 of size 300
Process 1 of size 417 is allocated in block 3 of size 500
Process 2 of size 112 is allocated in block 1 of size 200
Process 3 of size 426 is allocated in block 4 of size 600

Enter your choice: 3

--- Worst Fit ---
After sorting block sizes (Worst Fit):
Block 0: 100
Block 1: 0
Block 2: 0
Block 3: 0
Block 4: 0

Process 0 of size 212 is not allocated.
Process 1 of size 417 is not allocated.
Process 2 of size 112 is not allocated.
Process 3 of size 426 is not allocated.
```