Code 1:

Best Fit

```
// C++ implementation of Best - Fit algorithm
#include<iostream>
using namespace std;
// Method to allocate memory to blocks as per Best fit algorithm
void bestFit(int blockSize[], int m, int processSize[], int n)
{
// Stores block id of the block allocated to a process
int allocation[n];
// Initially no block is assigned to any process
for (int i = 0; i < n; i++)
allocation[i] = -1;
// pick each process and find suitable blocks
// according to its size ad assign to it
for (int i = 0; i < n; i++)
{
// Find the best fit block for current process
int bestIdx = -1;
for (int j = 0; j < m; j++)
{
if (blockSize[j] >= processSize[i])
{
if (bestIdx == -1)
bestIdx = j;
else if (blockSize[bestIdx] > blockSize[j])
bestIdx = j;
}
}
// If we could find a block for current process
if (bestIdx != -1)
{
// allocate block j to p[i] process
allocation[i] = bestIdx;
// Reduce available memory in this block.
blockSize[bestIdx] -= processSize[i];
}
}
cout << "\nProcess No.\tProcess Size\tBlock no.\n";
for (int i = 0; i < n; i++)
{
cout << " " << i+1 << "\t\t" << processSize[i] << "\t\t";
```

```
if (allocation[i] != -1)
cout << allocation[i] + 1;
else
cout << "Not Allocated";
cout << endl;
}
}
// Driver Method
int main()
{
int blockSize[] = {100, 500, 200, 300, 600};
int processSize[] = {212, 417, 112, 426};
int m = sizeof(blockSize) / sizeof(blockSize[0]);
int n = sizeof(processSize) / sizeof(processSize[0]);
bestFit(blockSize, m, processSize, n);
}
```

Output :

```
PS C:\Users\dgs\Desktop\OS OTT\os 1 vivek\OS 4 TO 10> cd "c:
RunnerFile } ; if ($?) { .\tempCodeRunnerFile }

Process No.     Process Size    Block no.
 1              212             4
 2              417             2
 3              112             3
 4              426             5
PS C:\Users\dgs\Desktop\OS OTT\os 1 vivek\OS 4 TO 10>
```

Code 2:

Worst Fit

```
#include<stdio.h>
#include<cstring>
#include<iostream>
using namespace std;
// Function to allocate memory to blocks as per worst fit
// algorithm
void worstFit(int blockSize[], int m, int processSize[],int n)
{
// Stores block id of the block allocated to a
// process
int allocation[n];
// Initially no block is assigned to any process
memset(allocation, -1, sizeof(allocation));
// pick each process and find suitable blocks
// according to its size ad assign to it
for (int i=0; i<n; i++)
{
// Find the best fit block for current process
```

```
int wstIdx = -1;
for (int j=0; j<m; j++)
{
if (blockSize[j] >= processSize[i])
{
if (wstIdx == -1)
wstIdx = j;
else if (blockSize[wstIdx] < blockSize[j])
wstIdx = j;
}}
// If we could find a block for current process
if (wstIdx != -1)
{
// allocate block j to p[i] process
allocation[i] = wstIdx;
// Reduce available memory in this block.
blockSize[wstIdx] -= processSize[i];
}}
cout << "\nProcess No.\tProcess Size\tBlock no.\n";
for (int i = 0; i < n; i++)
{
cout << " " << i+1 << "\t\t" << processSize[i] << "\t\t";
if (allocation[i] != -1)
cout << allocation[i] + 1;
else
cout << "Not Allocated";
cout << endl;
}}
// Driver code
int main()
{
int blockSize[] = {100, 500, 200, 300, 600};
int processSize[] = {212, 417, 112, 426};
int m = sizeof(blockSize)/sizeof(blockSize[0]);
int n = sizeof(processSize)/sizeof(processSize[0]);
worstFit(blockSize, m, processSize, n);
return 0 ;
}
```

Output :-

```
PS C:\Users\dgs\Desktop\OS OTT\os 1 vivek\OS 4 TO 10>
deRunnerFile } ; if ($?) { .\tempCodeRunnerFile }

Process No.     Process Size    Block no.
 1              212             5
 2              417             2
 3              112             5
 4              426             Not Allocated
PS C:\Users\dgs\Desktop\OS OTT\os 1 vivek\OS 4 TO 10>
```

Code 3:

First Fit :

```
#include<stdio.h>
void main()
{
int bsize[10], psize[10], bno, pno, flags[10], allocation[10], i, j;
for(i = 0; i < 10; i++)
{
flags[i] = 0;
allocation[i] = -1;
}
printf("Enter no. of blocks: ");
scanf("%d", &bno);
printf("\nEnter size of each block: ");
for(i = 0; i < bno; i++)
scanf("%d", &bsize[i]);
printf("\nEnter no. of processes: ");
scanf("%d", &pno);
printf("\nEnter size of each process: ");
for(i = 0; i < pno; i++)
scanf("%d", &psize[i]);
for(i = 0; i < pno; i++) //allocation as per first fit
for(j = 0; j < bno; j++)
if(flags[j] == 0 && bsize[j] >= psize[i])
{
allocation[j] = i;
flags[j] = 1;
break;
}
//display allocation details
printf("\nBlock no.\tsize\t\tprocess no.\t\tsize");
for(i = 0; i < bno; i++)
{
printf("\n%d\t\t%d\t\t", i+1, bsize[i]);
if(flags[i] == 1)
printf("%d\t\t\t%d",allocation[i]+1,psize[allocation[i]]);
else
printf("Not allocated");
}
```

Output:

```
PS C:\Users\dgs\Desktop\OS OTT\os 1 vivek\OS 4 TO 10> cd "c:\Users
RunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter no. of blocks: 3

Enter size of each block: 2
4
5

Enter no. of processes: 4

Enter size of each process: 1
3
4
6

Block no.       size            process no.             size
1               2               1                       1
2               4               2                       3
3               5               3                       4
PS C:\Users\dgs\Desktop\OS OTT\os 1 vivek\OS 4 TO 10>
```