EXP 4: BEST FIT-NIYATI SAVANT

Code:

```
#include<stdio.h>
#include<string.h>

struct Memory_partition
{
   int index;
   int msize;
   int mstatus;
   int mprocess;
}mem[10];

int main()
{
   int i,n,t,new_size,new_p,best_index,min_waste,x;
   n=8;
   t=9; //array size
   printf("\nVariable Partitioning- Best Fit-Niyati Savant");
   printf("\nThe Memory has 8 partitions");
   printf("\n Enter memory size status(0-Free and 1 for Not free)");
   for(i=0;i<n;i++)
   {
      mem[i].index=i;
      printf("\nEnter details for partition %d: ",i+1);
      printf("\nEnter Memory Size and Memory Status: ");
      scanf("%d %d",&mem[i].msize,&mem[i].mstatus);
      if(mem[i].mstatus==1)
      {
         printf("\nEnter process number:");
         scanf("%d",&mem[i].mprocess);
```

```
    }
    else
        mem[i].mprocess=0;
}

printf("\n Sr.No  Memory \t Status   Process \n");
for(i=0;i<n;i++)
{
    if(mem[i].mstatus==1)
    printf("%d  \t %d kb   Not Free   P%d\n",mem[i].index+1,mem[i].msize,mem[i].mprocess);
    else
    printf("%d \t %d kb   Free    \n",mem[i].index+1,mem[i].msize);
}
printf("\nEnter new process number: ");
scanf("%d",&new_p);
printf("\nEnter new process Size: ");
scanf("%d",&new_size);
min_waste=2000;
best_index=0;

for(i=0;i<n;i++)
{
    if(mem[i].mstatus==0) // Free
    {
        x=mem[i].msize-new_size;
        if((mem[i].msize>=new_size) && (min_waste >= x))
        {
            min_waste =x;
            best_index=mem[i].index;
        }
    }
```

```
}
printf("\n Best index %d",best_index+1);
printf("\n External fragmentation is %d",min_waste);

//Updating the mem statues
i=t-1;
   while((i-1) != best_index)
   {
      mem[i].index=i;
      mem[i].msize=mem[i-1].msize;
      mem[i].mstatus=mem[i-1].mstatus;
      if(mem[i].mstatus==1)
         mem[i].mprocess=mem[i-1].mprocess;
      i--;
   }
   mem[i].index=i;
   mem[i].msize=min_waste;
   mem[i].mstatus=0;
   i--;
   mem[i].index=best_index;
   mem[i].msize=new_size;
   mem[i].mstatus=1;
   mem[i].mprocess=new_p;


printf("\n After Updating");
printf("\n Sr.No  Memory \t Status   Process \n");
for(i=0;i<t;i++)
{
   if(mem[i].mstatus==1)
   printf("%d  \t %d kb   Not Free   P%d\n",i+1,mem[i].msize,mem[i].mprocess);
```

```
        else
        printf("%d \t %d kb   Free    \n",i+1,mem[i].msize);
    }


    return 0;
}
```

Output:

```
Variable Partitioning- Best Fit -Niyati Savant
The Memory has 8 partitions
 Enter memory size status(0-Free and 1 for Not free)
 Enter details for partition 1:
Enter Memory Size and Memory Status: 100 1
Enter process number:1
Enter details for partition 2:
Enter Memory Size and Memory Status: 300 0
Enter details for partition 3:
Enter Memory Size and Memory Status: 40 1
Enter process number:2
Enter details for partition 4:
Enter Memory Size and Memory Status: 50 0
Enter details for partition 5:
Enter Memory Size and Memory Status: 150 1
Enter process number:3
Enter details for partition 6:
Enter Memory Size and Memory Status: 240 0
Enter details for partition 7:
Enter Memory Size and Memory Status: 200 1
Enter process number:4
Enter details for partition 8:
Enter Memory Size and Memory Status: 400 0
```

Sr.No Memory Status Process

1 100 kb Not Free P1

2 300 kb Free

3 40 kb Not Free P2

4 50 kb Free

5 150 kb Not Free P3

6 240 kb Free

7 200 kb Not Free P4

8 400 kb Free

Enter new process number: 5

Enter new process Size: 200

Best index 6

External fragmentation is 40

After Updating

Sr.No Memory Status Process

1 100 kb Not Free P1

2 300 kb Free

3 40 kb Not Free P2

4 50 kb Free

5 150 kb Not Free P3

6 200 kb Not Free P5

7 40 kb Free

8 200 kb Not Free P4

9 400 kb Free