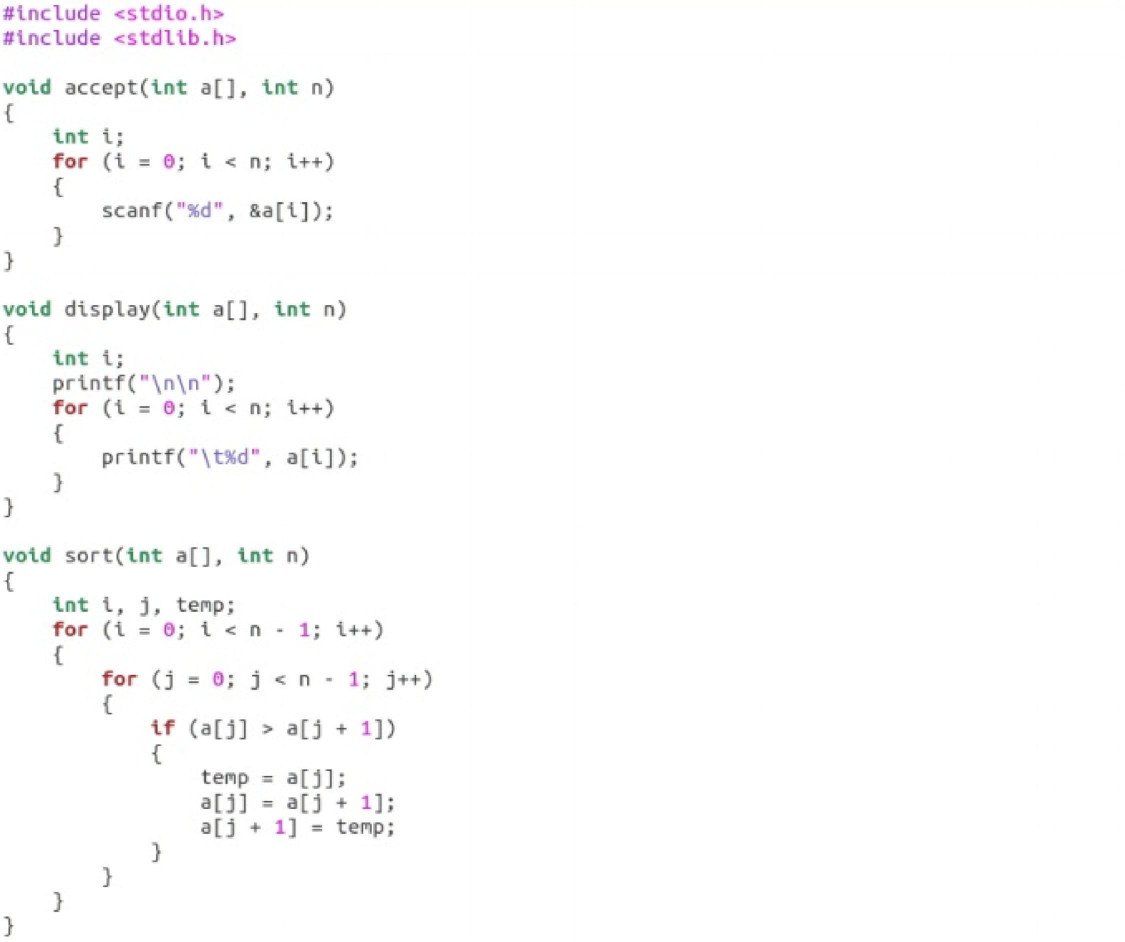
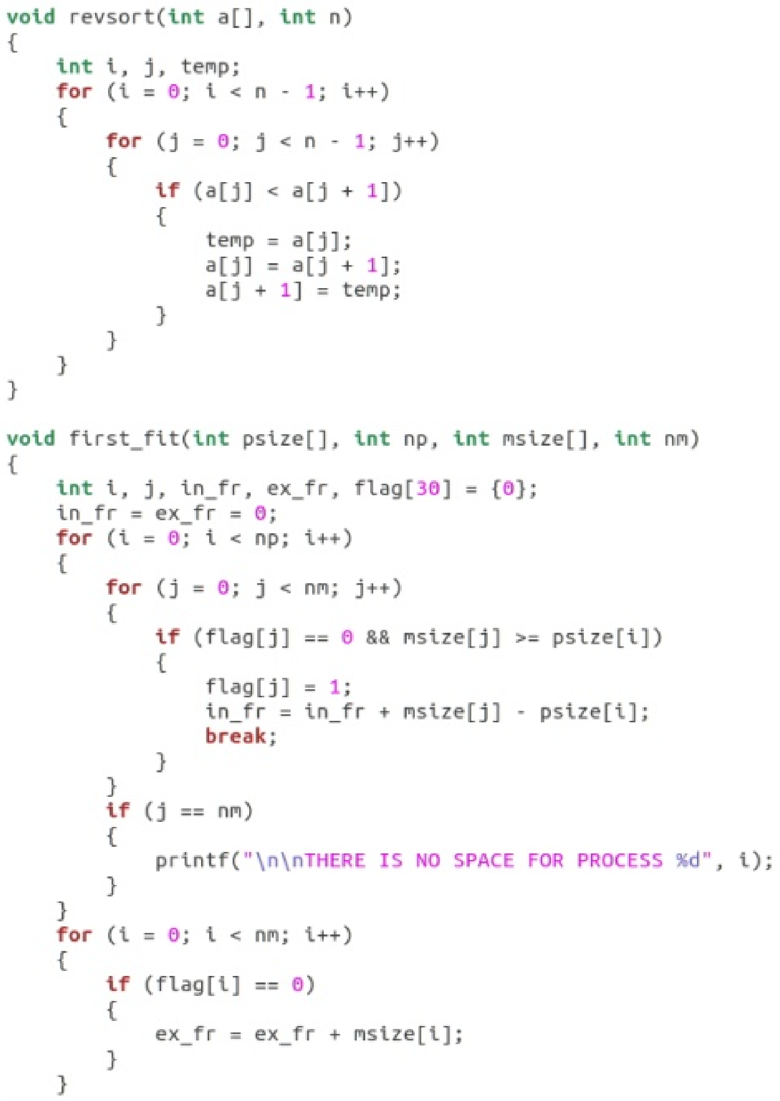
# 12Q. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

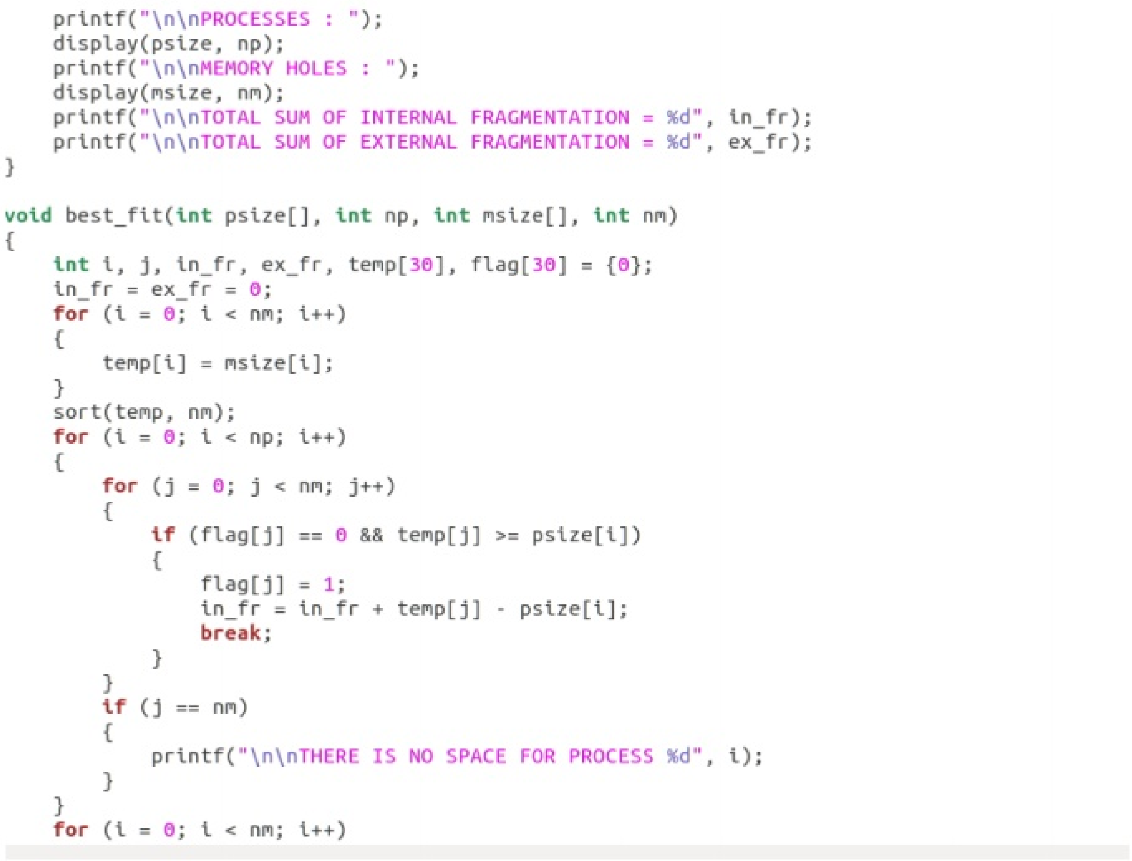
Code:



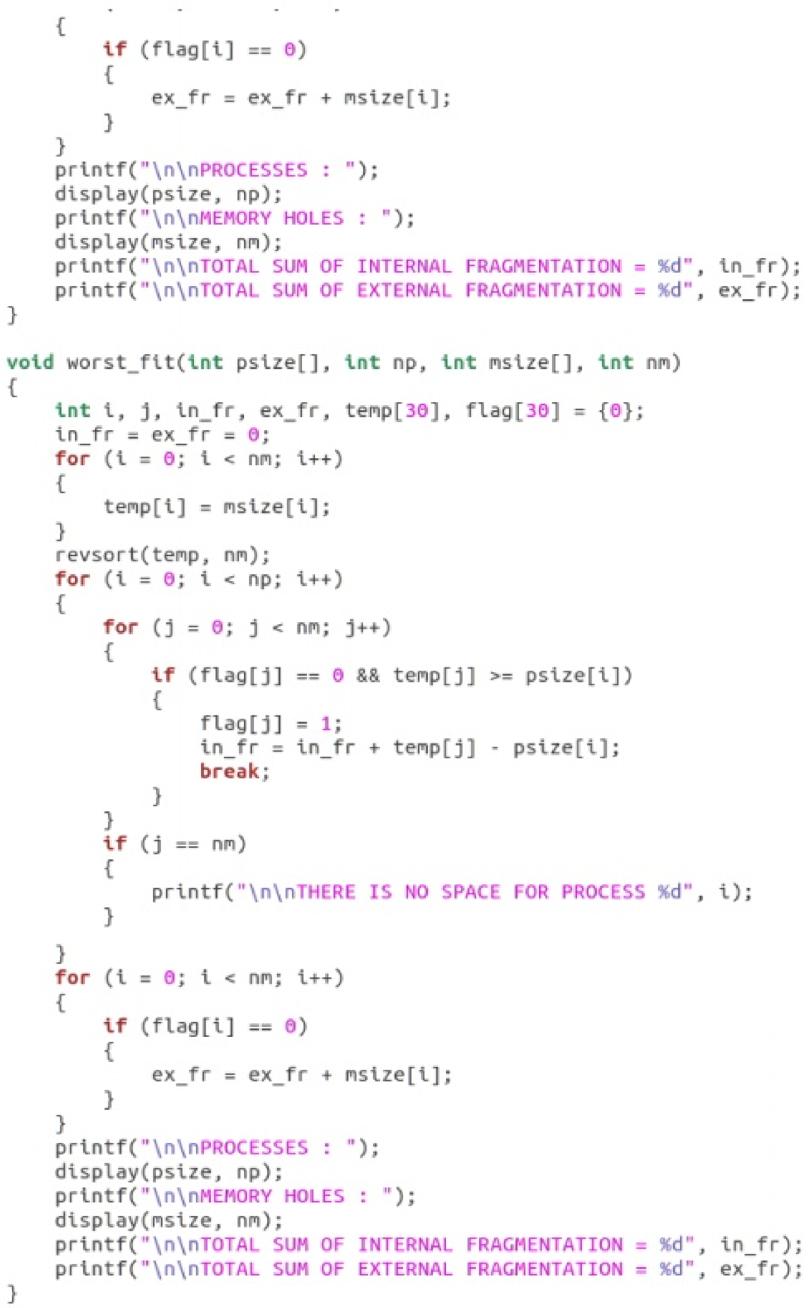
```c
#include <stdio.h>
#include <stdlib.h>

void accept(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
}

void display(int a[], int n)
{
    int i;
    printf("\n\n");
    for (i = 0; i < n; i++)
    {
        printf("\t%d", a[i]);
    }
}

void sort(int a[], int n)
{
    int i, j, temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = 0; j < n - 1; j++)
        {
            if (a[j] > a[j + 1])
            {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }
}
```



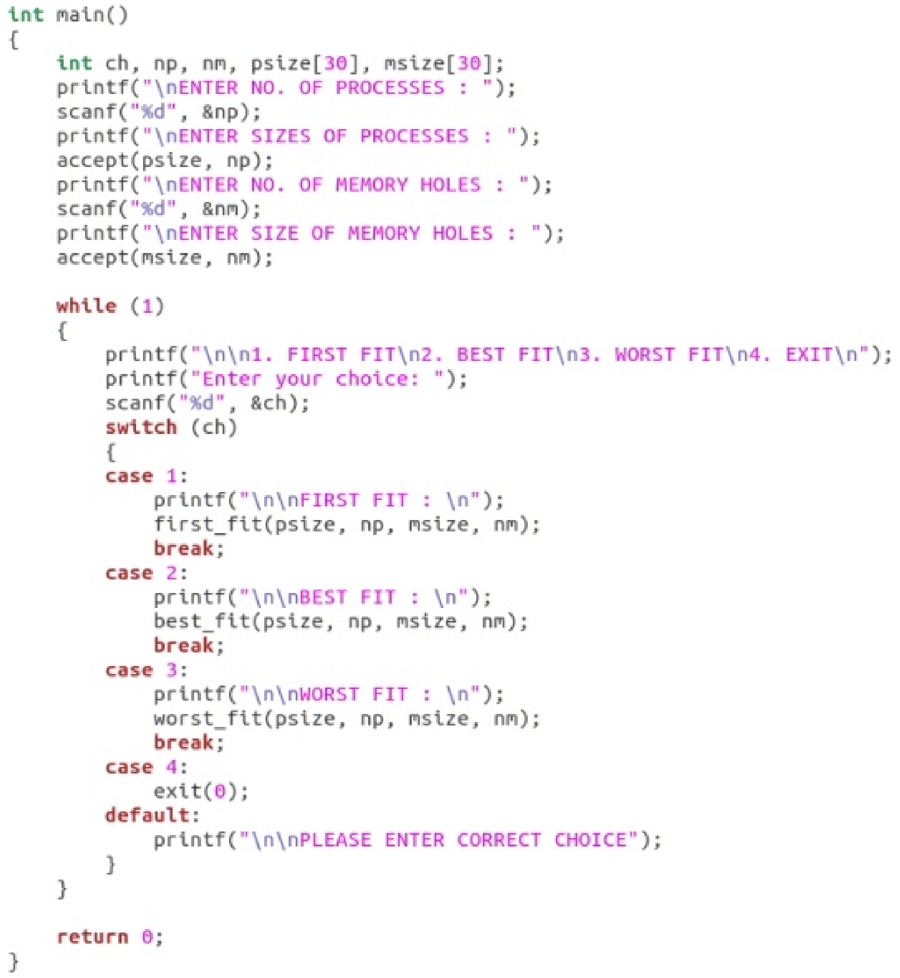
```
void revsort(int a[], int n)
{
    int i, j, temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = 0; j < n - 1; j++)
        {
            if (a[j] < a[j + 1])
            {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }
}

void first_fit(int psize[], int np, int msize[], int nm)
{
    int i, j, in_fr, ex_fr, flag[30] = {0};
    in_fr = ex_fr = 0;
    for (i = 0; i < np; i++)
    {
        for (j = 0; j < nm; j++)
        {
            if (flag[j] == 0 && msize[j] >= psize[i])
            {
                flag[j] = 1;
                in_fr = in_fr + msize[j] - psize[i];
                break;
            }
        }
        if (j == nm)
        {
            printf("\n\nTHERE IS NO SPACE FOR PROCESS %d", i);
        }
    }
    for (i = 0; i < nm; i++)
    {
        if (flag[i] == 0)
        {
            ex_fr = ex_fr + msize[i];
        }
    }
```



```
    printf("\n\nPROCESSES : ");
    display(psize, np);
    printf("\n\nMEMORY HOLES : ");
    display(msize, nm);
    printf("\n\nTOTAL SUM OF INTERNAL FRAGMENTATION = %d", in_fr);
    printf("\n\nTOTAL SUM OF EXTERNAL FRAGMENTATION = %d", ex_fr);
}

void best_fit(int psize[], int np, int msize[], int nm)
{
    int i, j, in_fr, ex_fr, temp[30], flag[30] = {0};
    in_fr = ex_fr = 0;
    for (i = 0; i < nm; i++)
    {
        temp[i] = msize[i];
    }
    sort(temp, nm);
    for (i = 0; i < np; i++)
    {
        for (j = 0; j < nm; j++)
        {
            if (flag[j] == 0 && temp[j] >= psize[i])
            {
                flag[j] = 1;
                in_fr = in_fr + temp[j] - psize[i];
                break;
            }
        }
        if (j == nm)
        {
            printf("\n\nTHERE IS NO SPACE FOR PROCESS %d", i);
        }
    }
    for (i = 0; i < nm; i++)
```

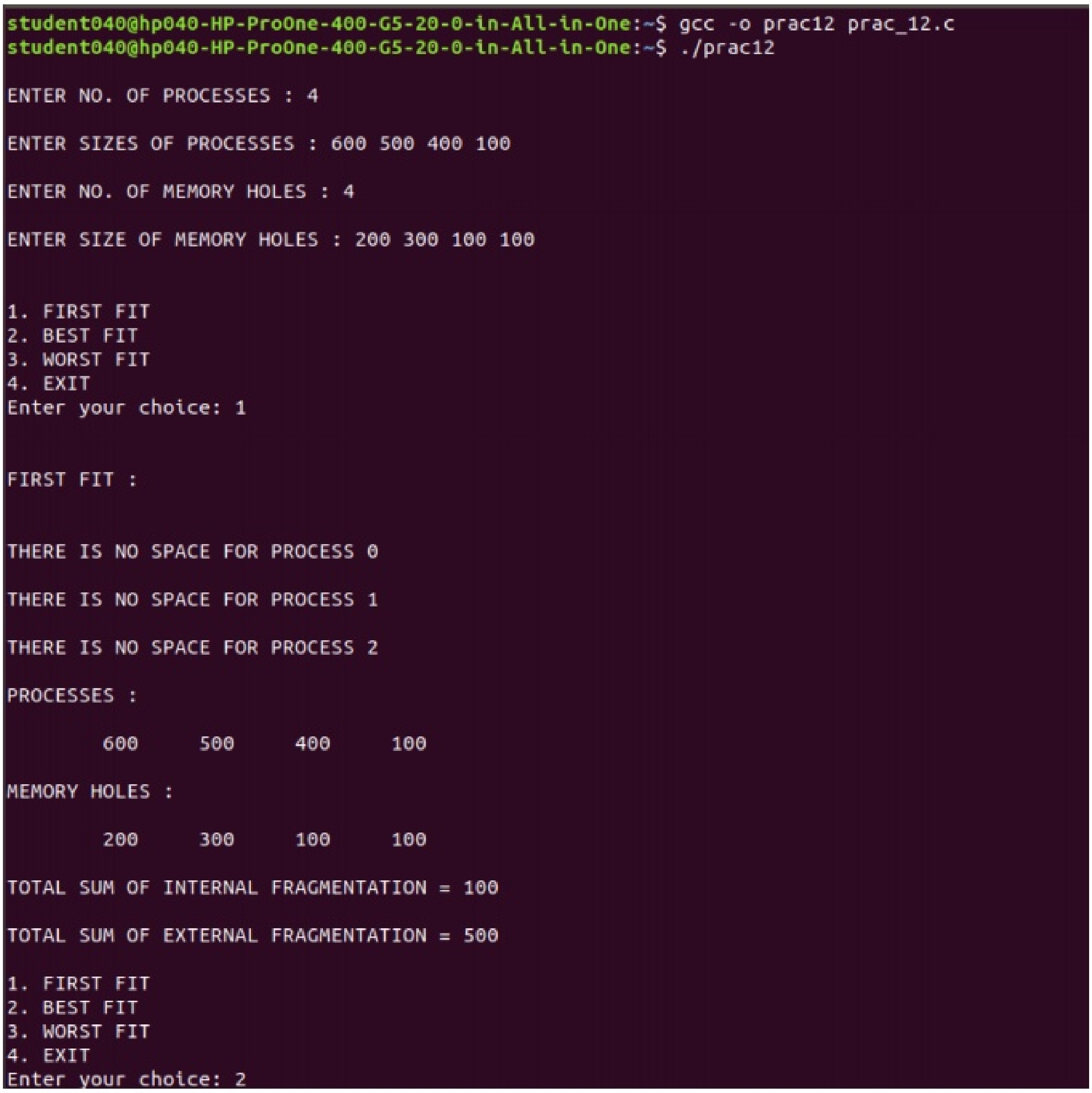


```
    {
        if (flag[i] == 0)
        {
            ex_fr = ex_fr + msize[i];
        }
    }
    printf("\n\nPROCESSES : ");
    display(psize, np);
    printf("\n\nMEMORY HOLES : ");
    display(msize, nm);
    printf("\n\nTOTAL SUM OF INTERNAL FRAGMENTATION = %d", in_fr);
    printf("\n\nTOTAL SUM OF EXTERNAL FRAGMENTATION = %d", ex_fr);
}

void worst_fit(int psize[], int np, int msize[], int nm)
{
    int i, j, in_fr, ex_fr, temp[30], flag[30] = {0};
    in_fr = ex_fr = 0;
    for (i = 0; i < nm; i++)
    {
        temp[i] = msize[i];
    }
    revsort(temp, nm);
    for (i = 0; i < np; i++)
    {
        for (j = 0; j < nm; j++)
        {
            if (flag[j] == 0 && temp[j] >= psize[i])
            {
                flag[j] = 1;
                in_fr = in_fr + temp[j] - psize[i];
                break;
            }
        }
        if (j == nm)
        {
            printf("\n\nTHERE IS NO SPACE FOR PROCESS %d", i);
        }

    }
    for (i = 0; i < nm; i++)
    {
        if (flag[i] == 0)
        {
            ex_fr = ex_fr + msize[i];
        }
    }
    printf("\n\nPROCESSES : ");
    display(psize, np);
    printf("\n\nMEMORY HOLES : ");
    display(msize, nm);
    printf("\n\nTOTAL SUM OF INTERNAL FRAGMENTATION = %d", in_fr);
    printf("\n\nTOTAL SUM OF EXTERNAL FRAGMENTATION = %d", ex_fr);
}
```

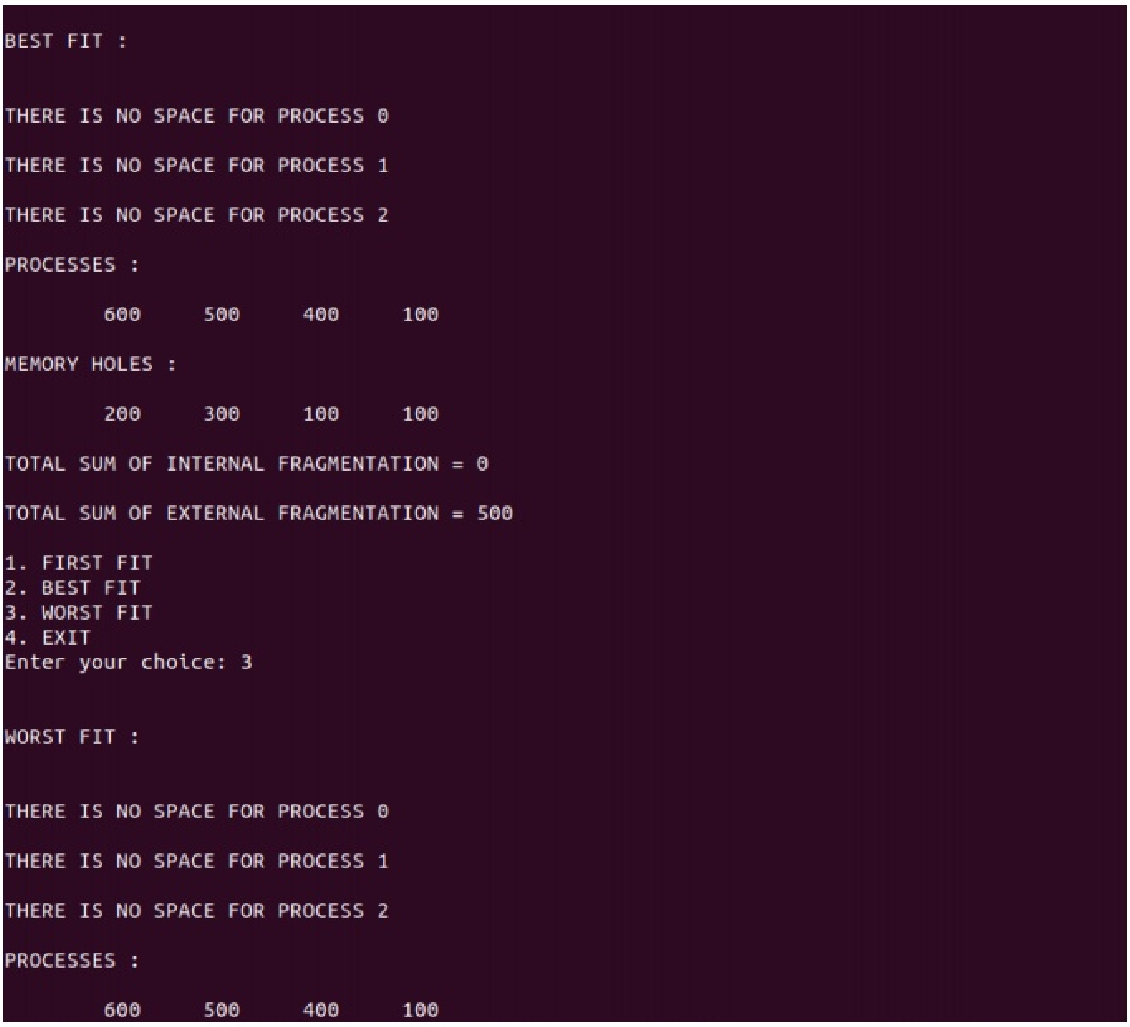


```c
int main()
{
    int ch, np, nm, psize[30], msize[30];
    printf("\nENTER NO. OF PROCESSES : ");
    scanf("%d", &np);
    printf("\nENTER SIZES OF PROCESSES : ");
    accept(psize, np);
    printf("\nENTER NO. OF MEMORY HOLES : ");
    scanf("%d", &nm);
    printf("\nENTER SIZE OF MEMORY HOLES : ");
    accept(msize, nm);

    while (1)
    {
        printf("\n\n1. FIRST FIT\n2. BEST FIT\n3. WORST FIT\n4. EXIT\n");
        printf("Enter your choice: ");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            printf("\n\nFIRST FIT : \n");
            first_fit(psize, np, msize, nm);
            break;
        case 2:
            printf("\n\nBEST FIT : \n");
            best_fit(psize, np, msize, nm);
            break;
        case 3:
            printf("\n\nWORST FIT : \n");
            worst_fit(psize, np, msize, nm);
            break;
        case 4:
            exit(0);
        default:
            printf("\n\nPLEASE ENTER CORRECT CHOICE");
        }
    }

    return 0;
}
```

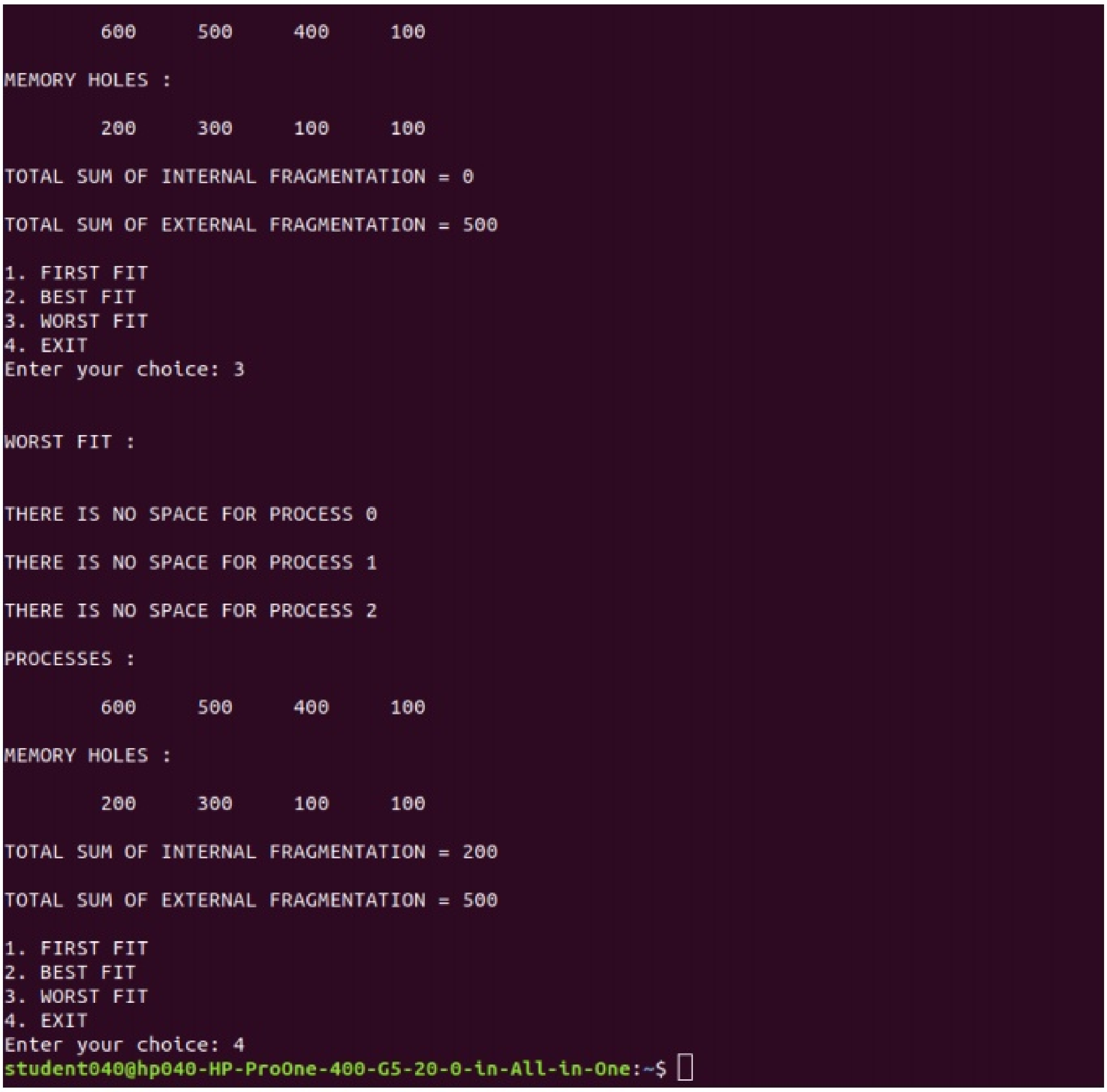
## Output:



```
student040@hp040-HP-ProOne-400-G5-20-0-in-All-in-One:~$ gcc -o prac12 prac_12.c
student040@hp040-HP-ProOne-400-G5-20-0-in-All-in-One:~$ ./prac12

ENTER NO. OF PROCESSES : 4

ENTER SIZES OF PROCESSES : 600 500 400 100

ENTER NO. OF MEMORY HOLES : 4

ENTER SIZE OF MEMORY HOLES : 200 300 100 100


1. FIRST FIT
2. BEST FIT
3. WORST FIT
4. EXIT
Enter your choice: 1


FIRST FIT :


THERE IS NO SPACE FOR PROCESS 0

THERE IS NO SPACE FOR PROCESS 1

THERE IS NO SPACE FOR PROCESS 2

PROCESSES :

        600     500     400     100

MEMORY HOLES :

        200     300     100     100

TOTAL SUM OF INTERNAL FRAGMENTATION = 100

TOTAL SUM OF EXTERNAL FRAGMENTATION = 500

1. FIRST FIT
2. BEST FIT
3. WORST FIT
4. EXIT
Enter your choice: 2
```



```
BEST FIT :


THERE IS NO SPACE FOR PROCESS 0

THERE IS NO SPACE FOR PROCESS 1

THERE IS NO SPACE FOR PROCESS 2

PROCESSES :

        600     500     400     100

MEMORY HOLES :

        200     300     100     100

TOTAL SUM OF INTERNAL FRAGMENTATION = 0

TOTAL SUM OF EXTERNAL FRAGMENTATION = 500

1. FIRST FIT
2. BEST FIT
3. WORST FIT
4. EXIT
Enter your choice: 3


WORST FIT :


THERE IS NO SPACE FOR PROCESS 0

THERE IS NO SPACE FOR PROCESS 1

THERE IS NO SPACE FOR PROCESS 2

PROCESSES :

        600     500     400     100
```



```
        600     500     400     100

MEMORY HOLES :

        200     300     100     100

TOTAL SUM OF INTERNAL FRAGMENTATION = 0

TOTAL SUM OF EXTERNAL FRAGMENTATION = 500

1. FIRST FIT
2. BEST FIT
3. WORST FIT
4. EXIT
Enter your choice: 3


WORST FIT :


THERE IS NO SPACE FOR PROCESS 0

THERE IS NO SPACE FOR PROCESS 1

THERE IS NO SPACE FOR PROCESS 2

PROCESSES :

        600     500     400     100

MEMORY HOLES :

        200     300     100     100

TOTAL SUM OF INTERNAL FRAGMENTATION = 200

TOTAL SUM OF EXTERNAL FRAGMENTATION = 500

1. FIRST FIT
2. BEST FIT
3. WORST FIT
4. EXIT
Enter your choice: 4
student040@hp040-HP-ProOne-400-G5-20-0-in-All-in-One:~$
```