

## I. REQUIREMENTS

Develop a Java program that implements three memory allocation methods:

- **First-Fit (FF)**
- **Best-Fit (BF)**
- **Worst-Fit (WF)**

The program will INPUT

- reading data on available memory blocks from `Minput.data` and
- process requirements from `Pinput.data`

It should then allocate memory using the three methods and OUTPUT results in

- `FFoutput.data`,
- `BFoutput.data`,
- `WFoutput.data`.

Please see the formats for these 5 files below.

Then, write a report about this programming project and submit both working folder (including codes and data files) and the report.

## II. FORMATS

Attention: whatever inside () is only for explanation and should not be present/printed in the data files.

Format for “Minput.data”:

3 (#of free memory slots)  
100 400 (addresses of start and end of a free memory slot => size 300)  
600 800 (addresses of start and end of a free memory slot => size 200)  
1500 1900 (addresses of start and end of a free memory slot => size 400)

Format for “Pinput.data”:

3 (# of processes)  
1 (ID of process) 190 (size of process)  
2 (ID of process) 210 (size of process)  
3 (ID of process) 205 (size of process)

Format for “FFoutput.data”, “BFoutput.data”, and “WFoutput.data”

100(addresses of start) 310(addresses of end) 2(process ID)  
600(addresses of start) 790(addresses of end) 1(process ID)  
1500(addresses of start) 1705(addresses of end) 3(process ID)  
-0 (means all are allocated, or -1,3 if processes 1 and 3 can't be allocated)

Format of the report of the project:

**Title Page:** Student name and memory allocation methods used.

### **Table of Contents**

1. **Algorithm Descriptions:** Explain each method with an example.
2. **Implementation Details:** Describe your program.
3. **Experiments:**
  - Run your program with two different input datasets.
  - Provide results for FF, BF, and WF for each dataset.
  - Include charts for each method.
4. **Conclusions**
5. **References**

### **III. WHAT TO SUBMIT**

- 1) Report file in word format (named as “LastName-FirstInitial.doc”) separately from
- 2) ZIP/RAR file (named as “LastName-FirstInitial.zip/rar”) which contains a folder with at least the following files:
  - Minput.data, Pinput.data
  - FFoutput.data, BFoutput.data, and WFoutput.data
  - Mallocator.class (executable code)
  - Mallocator.java (source code)

**WHERE TO SUBMIT:** Brightspace

#### **DEADLINE:**

- (A) Codes (working folder): ZIP/RAR file (named as “LastName-FirstInitial.zip/rar”, see above) is due before **11:00pm of the Sunday** of the week when we have the **1st lab (A)** for this PROJECT.
- (B) Report: DOCX or PDF file for the report is due before **11:00pm of the Sunday** of the week when we have the **2nd lab (B)** for this PROJECT. Please also **resubmit the codes (A)**.

**NOTE:** For each lab based on this project, at the end of each lab: submit a zipped file of your working folder which includes everything including the report file, to show your progress. For the project itself final version of report, software and data must be submitted (11:55pm of the day of the last lab based on this project).

### **IV. EXAMPLES**

Please find below an example for the **EXPERIMENTS** part of the report.

### 3. EXPERIMENTS:

#### Experiment #1:

##### a) INPUT Data:

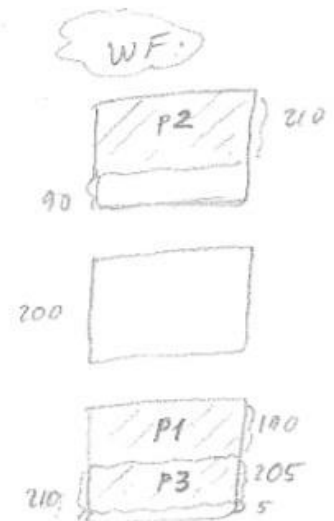
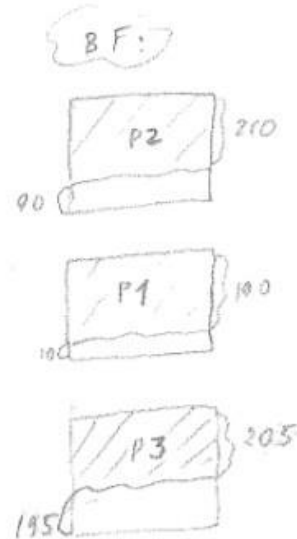
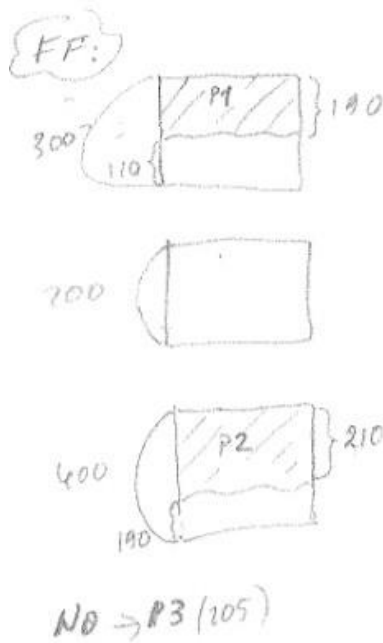
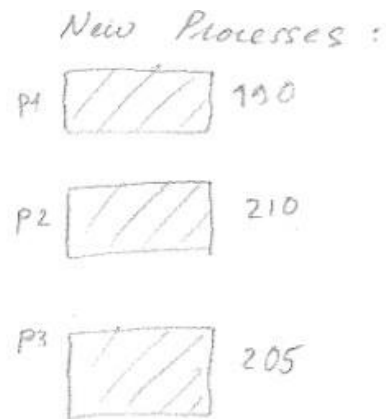
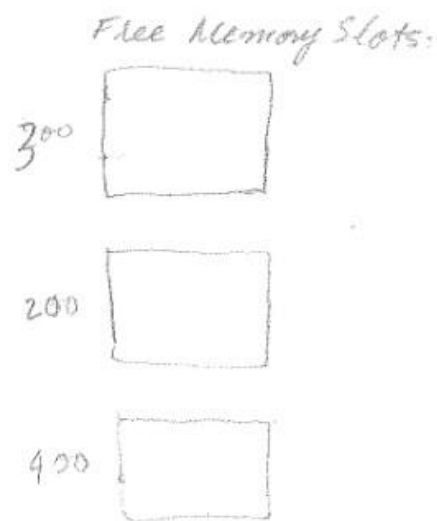
"Minput.data"

3  
100 400  
600 800  
1500 1900

"Pinput.data"

3  
1 190  
2 210  
3 205

##### b) Charts :



**c) SHOULD-BE Output:**

```
"FFoutput.data"  
100 290 1  
1500 1710 2  
-3
```

```
"BFoutput.data"  
100 310 2  
600 790 1  
1500 1705 3  
-0
```

```
"WFoutput.data"  
100 310 2  
1500 1690 1  
1690 1895 3  
-0
```

**d) MY Output (produced by my program):**

```
"FFoutput.data"  
100 290 1  
1500 1710 2  
-3
```

```
"BFoutput.data"  
100 310 2  
600 790 1  
1500 1705 3  
-0
```

```
"WFoutput.data"  
100 310 2  
1500 1690 1  
1690 1895 3  
-0
```