**Q1**

public static void main(String[] args) {

for (int i = - 0 ; i < args.length; i++)

{

System.out.println("Value of input is" + i + "and arguments" + args[i]);

}

}

**Q2**

The system will throw AccessControlException: access denied exception, since the given program is trying to create a new file.

**Q3**

public static void outOfMemoryError(List<String> records) {

StringBuffer sb = new StringBuffer();

try {

while (true) {

sb.append("testtest");

}

} catch (OutOfMemoryError ex) {

System.out.println("OutOfMemoryError Occurred!!");

System.out.println(ex.getMessage());

}

}

public static String stringBufferWithoutGC(List<String> records) {

if (records == null) {

return null;

}

if (records.isEmpty()) {

return "";

}

int expSize = 0;

for (String record : records) {

expSize += record.length();

}

StringBuffer sb = new StringBuffer(expSize);

for (String record : records) {

sb.append(record);

}

return sb.toString();

}

To use with GC add -XX:+HeapDumpOnOutOfMemoryError in argument of program

Q4

public class MenuItem {

private String category;

private String name;

private int price;

public MenuItem(String category, String name, int price) {

this.category = category;

this.name = name;

this.price = price;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

@Override

public String toString() {

return "MenuItem{" + "category='" + category + '\'' + ", name='" + name + '\'' + ", price=" + price + '}';

}

}

public static void main(String[] args) {

MenuItem item1 = new MenuItem("fast food", "Burger", 100);

MenuItem item2 = new MenuItem("fast food", "Zinger", 200);

MenuItem item3 = new MenuItem("barbeque", "Kebab", 600);

MenuItem item4 = new MenuItem("barbeque", "Grilled Chicken", 400);

MenuItem item5 = new MenuItem("barbeque", "Reshmi Kebab", 500);

List<MenuItem> items = new ArrayList<>();

items.add(item1);

items.add(item2);

items.add(item3);

items.add(item4);

items.add(item5);

readMenuItem(items);

}

public static void readMenuItem(List<MenuItem> items)

{

items.sort(Comparator.comparing(MenuItem::getCategory).thenComparing(MenuItem::getName).thenComparing(MenuItem::getPrice));

System.out.println(items);

}