**JAVA**

Ques 1-

Write a code to print the following patterns:

    \*

   \* \*

  \* \* \*

 \* \* \* \*

\* \* \* \* \*

        A

      B C

    D E F

  G H I J

K L M N O

**CODE**

public class a {

public static void main(String[] args) {

int rows = 5;

char temp='A';

for (int i = 1; i <= rows; ++i) {

for (int j = 1; j <= rows; ++j) {

if(i+j>5){

System.out.print(temp);

temp++;

}

else{

System.out.print(" ");

}

}

System.out.println();

}

for (int i=0; i<rows; i++)

{

for (int j=rows-i; j>1; j--)

{

// printing spaces

System.out.print(" ");

}

**//for the second pattern**

for (int j=0; j<=i; j++ )

{

// printing stars

System.out.print("\* ");

}

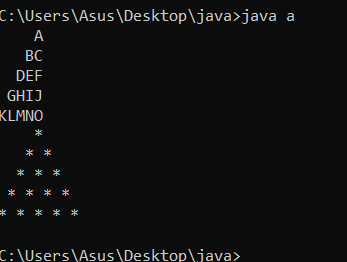
// ending line after each row

System.out.println();

}

}

}



Ques 2-

Problem Statement :

Make a Grocery List for supermarket shopping with name, price and quantity; if the list already contains an item then only update the price and quantity it should not append the item name again. Ask the user his/her budget initially and minus the budget after adding a new item in the list. If budgets go zero/0 then no more items could be bought and if some money left and users add items greater than budget left then inform “over price” or any other message. After the list is made any money left in the budget it should show an item within the budget from the list made.VALIDATION is a must.

Examples:

User GO with following question

Enter Your budget : 500

1.Add an item

2.Exit

Enter your choice : 1

Enter product : corn flour

Enter quantity : 1.5 kg

Enter Price : 100

Amount left : 400

1.Add an item

2.Exit

Enter your choice : 1

Enter product : wheat

Enter quantity : 2 kg

Enter Price : 100

Amount left : 300

1.Add an item

2.Exit

Enter your choice : 1

Enter product : corn flour

Enter quantity : 2 kg

Enter Price : 250

Amount left : 150

1.Add an item

2.Exit

Enter your choice : 1

Enter product : rice

Enter quantity : 5 kg

Enter Price : 300

Can't Buy the product ###(because budget left is 150)

1.Add an item

2.Exit

Enter your choice : 1

Enter product : xyz

Enter quantity : 1 kg

Enter Price : 50

Amount left : 100

1.Add an item

2.Exit

Enter your choice : 2

Amount left can buy you wheat

GROCERY LIST is:

Product name   Quantity   Price

corn flour      2 kg        250

wheat           2 kg        100

xyz             1 kg         50

**CODE:**

import java.util.\*;

public class Main {

public static void main(String[] args) {

boolean b=true;

ArrayList<Integer> price = new ArrayList<Integer>();

ArrayList<String> quantity = new ArrayList<String>();

ArrayList<String> product = new ArrayList<String>();

int i=0;

System.out.print("Enter your budget:");

java.util.Scanner sc=new java.util.Scanner(System.in);

int budget=sc.nextInt();

while(b){

System.out.print("1.Add an item\n2.Exit\nEnter your choice :");

int choice=sc.nextInt();

switch(choice){

case 1:

System.out.print("Enter product :");

String pro=sc.next();

System.out.print("Enter quantity :");

String qua=sc.next();

System.out.print("Enter Price :");

int pri=sc.nextInt();

if(pri<=budget){

System.out.println("Amount left : "+(budget-pri));

price.add(pri);

product.add(pro);

quantity.add(qua);

budget=budget-pri;

}

else{

System.out.println("Can't Buy the product ###(because budget left is"+budget);

}

break;

case 2:

if(budget>0){

for(i=0;i<price.size();i++){

if(budget>price.get(i)){

System.out.println("Amount left can buy you "+product.get(i));

break;

}

}

}

System.out.println("Product name Quantity Price");

for(i=0;i<price.size();i++){

System.out.printf(product.get(i));

System.out.printf("%15s",quantity.get(i));

System.out.print(price.get(i)+"\n");

}

b=false;

break;

default:

System.out.println("Wrong choice");

}

}

}

}

**ANDROID**

Challenge-

Create a functional mobile Calculator App.

Mandatory Features-

* Language used should be Java.
* Calculations should be performed for numbers having 3-digits.
* The numeric buttons should be rectangular with a curved radius of 40 px and the operator keys should be circular.

It should perform **arithmetic calculations** on Positive, Negative and decimal numbers.