**Question 1:**

**#!/bin/sh**

**#lastarg – prints out the last argument in the list**

**#creates a counter variable called i**

**i=1**

**#creates a variable numb\_args that is equal to the number of arguments**

**numb\_args=$#**

**#if no argument is provided (number of arguments is zero) return nothing**

**if [ $# -eq 0 ]**

**then**

**:**

**#otherwise script returns the number of arguments and the last argument**

**else**

**echo the number of arguments $#**

**echo "The last argument is:"**

**#while loop to shift all the arguments so that the last argument becomes the first**

**while [ $i -lt $numb\_args ]**

**do**

**#shifts the agruments**

**shift**

**#increments the value of the counter variable by 1**

**i=`expr $i + 1`**

**done**

**#prints out the argument at location 1**

**echo $1**

**fi**

**obelix.gaul.csd.uwo.ca[88]% lastarg**

**obelix.gaul.csd.uwo.ca[89]% lastarg a1 a2 a3 a4 a5**

**the number of arguments 5**

**The last argument is:**

**a5**

**obelix.gaul.csd.uwo.ca[90]% lastarg a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13**

**the number of arguments 13**

**The last argument is:**

**a13**

**obelix.gaul.csd.uwo.ca[91]% lastarg this is the last word**

**the number of arguments 5**

**The last argument is:**

**word**

**obelix.gaul.csd.uwo.ca[15]% lastarg**

**obelix.gaul.csd.uwo.ca[16]%**

*If lastarg is placed in your home directory, what will happen if you execute the following command? Show the output and explain why you got this output. cd; lastarg .\**

We are calling the **lastarg** shell script and passing all the files in the home directory that start with a **.** (dot). Thus the script reads (in lexicographical order) through all of the files that start with a dot and prints the last file name that starts with a dot. The last file that starts with a dot is named **.xsession-errors** and so the script prints this.

**obelix.gaul.csd.uwo.ca[82]% cd; lastarg .\***

**the number of arguments 37**

**The last argument is:**

**.xsession-errors**

**Question 2:**

**#!/bin/sh**

**#odd\_prn - echoes its shell script file name as well as its odd arguments**

**#creates a variable numb\_args that is equal to the number of arguments**

**numb\_args=$#**

**#creates a counter variable called i**

**i=1**

**#prints its own file name to the screen**

**echo "This is the shell script file name: " $0**

**#if no argument is provided (number of arguments is zero) return nothing**

**if [ $# -eq 0 ]**

**then**

**:**

**#otherwise script returns the number of arguments and the last argument**

**else**

**#echoes the number of arguments as well as the statement "Here are the odd arguments"**

**echo the number of arguments $#**

**echo "Here are the odd arguments:"**

**#prints the current argument at position $1 (this is the first argument before we enter the loop)**

**echo $1**

**#while loop to shift all the arguements so that odd arguments are shifted to position $1**

**while [ $i -lt $numb\_args ]**

**do**

**#if the crrent number of arguments is greater than 2**

**then**

**#shifts the agruments twice so that arguements at odd positions will shift to postion $1**

**if [ $# -gt 2 ]**

**then**

**shift**

**shift**

**#prints the current argument at position $1**

**echo $1**

**fi**

**#increments the value of the counter variable by 1**

**i=`expr $i + 1`**

**done**

**fi**

**obelix.gaul.csd.uwo.ca[120]% odd\_prn 1 2 3 4 5 6**

**This is the shell script file name: odd\_prn**

**the number of arguments 6**

**Here are the odd arguments:**

**1**

**3**

**5**

**obelix.gaul.csd.uwo.ca[121]% odd\_prn**

**This is the shell script file name: odd\_prn**

**obelix.gaul.csd.uwo.ca[122]% odd\_prn 1**

**This is the shell script file name: odd\_prn**

**the number of arguments 1**

**Here are the odd arguments:**

**1**

**obelix.gaul.csd.uwo.ca[123]% odd\_prn a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13**

**This is the shell script file name: odd\_prn**

**the number of arguments 13**

**Here are the odd arguments:**

**a1**

**a3**

**a5**

**a7**

**a9**

**a11**

**a13**

**obelix.gaul.csd.uwo.ca[124]% odd\_prn to C or not to C that is the question**

**This is the shell script file name: odd\_prn**

**the number of arguments 10**

**Here are the odd arguments:**

**to**

**or**

**to**

**that**

**the**

If odd\_prn is placed in your home directory, what will happen if you execute the following command? Show the output and explain why you got this output. cd; odd\_prn .\*

We are calling the **odd\_prn** shell script and passing all the files in the home directory that start with a **.** (dot). Thus the script reads through all of the files that start with a dot (in lexicographical order) and prints the odd arguments.

**obelix.gaul.csd.uwo.ca[127]% cd; odd\_prn .\***

**This is the shell script file name: odd\_prn**

**the number of arguments 37**

**Here are the odd arguments:**

**.**

**.ICEauthority**

**.Xauthority**

**.alias.rs6000**

**.alias.sun4m**

**.bash\_history**

**.config**

**.dbus**

**.dmrc.15-09-24**

**.gconf**

**.gnome2**

**.history.sun4**

**.local**

**.mwmrc**

**.plan**

**.profile**

**.ssh**

**.twmrc**

**.xsession-errors**

**Question 3:**

**#!/bin/sh**

**#number pyramid – creates a sideways pyramid of numbers starting with 0 and incrementing to the number the user inputs, and then decrements back to zero. If argument passed is not an integer**

**#initialize variables**

**i=0**

**j=0**

**#prompts user for input and stores in the variable user\_input**

**echo "Please enter an integer, or type "'"exit"'" to quit: "**

**read user\_input**

**#echoes back the user input and redirects output to grep. temp variable test**

**#if the grep conditions are met (temp equals 0 if user\_input is a valid integer)**

**echo "Your input is:"**

**echo "$user\_input" | grep '^[0-9]\*$'**

**temp=`echo "$?"`**

**#while the value the user inputs is not an integer it will loop and prompt**

**#the user again until the input is a proper integer**

**while [ $temp -ne 0 2> /dev/null ]**

**do**

**#if the user inputs the word "exit" then the script's exist status will**

**#become 0, terminating the script**

**if [ $user\_input = 'exit' ]**

**then**

**echo "exit"**

**exit 0**

**fi**

**echo "not a valid option."**

**#prompts user for input and stores in user\_input**

**echo "Please enter an integer, or type "'"exit"'" to quit: "**

**read user\_input**

**#echoes back the user input and redirects output to grep. temp variable test**

**#if the grep conditions are met (temp equals 0 if user\_input is a valid integer)**

**echo "Your input is: "**

**echo $user\_input | grep '^[0-9]\*$'**

**temp=`echo $?`**

**done**

**#nested loops to print half of the pyramid (incrementing from 0 to user\_input -1**

**#outer loop to iterate through rows**

**while [ $i -le $user\_input ]**

**do**

**#resets j to zero**

**j=0**

**#inner loop to iterate through columns within that row**

**while [ $j -lt $i ]**

**do**

**#print the current value of j and then a space**

**printf "$j "**

**#increments j by 1**

**j=`expr $j + 1`**

**done**

**#prints a new line**

**printf "\n"**

**#increments the current value of i by 1**

**i=`expr $i + 1`**

**done**

**#sets the value of i to be user\_input -2, which will be one less from where the top half**

**#of the pyramid ended**

**i=`expr $user\_input - 1`**

**#nested loops to print the rest of the pyramid**

**while [ $i -gt 0 ]**

**do**

**#resets j to zero**

**j=0**

**#inner loop to iterate through columns within that row**

**while [ $j -lt $i ]**

**do**

**#print the current value of j and then a space**

**printf "$j "**

**#increments j by 1**

**j=`expr $j + 1`**

**done**

**#echoes a new line**

**printf "\n"**

**#increments the current value of i by 1**

**i=`expr $i - 1`**

**done**

**obelix.gaul.csd.uwo.ca[45]% number\_pyramid**

**Please enter an integer, or type "exit" to quit:**

**wakemeupinside**

**Your input is:**

**not a valid option.**

**Please enter an integer, or type "exit" to quit:**

**-2**

**Your input is:**

**not a valid option.**

**Please enter an integer, or type "exit" to quit:**

**3.7**

**Your input is:**

**not a valid option.**

**Please enter an integer, or type "exit" to quit:**

**3dkjnk**

**Your input is:**

**not a valid option.**

**Please enter an integer, or type "exit" to quit:**

**3**

**Your input is:**

**3**

**0**

**0 1**

**0 1 2**

**0 1**

**0**

**obelix.gaul.csd.uwo.ca[46]% number\_pyramid**

**Please enter an integer, or type "exit" to quit:**

**6**

**Your input is:**

**6**

**0**

**0 1**

**0 1 2**

**0 1 2 3**

**0 1 2 3 4**

**0 1 2 3 4 5**

**0 1 2 3 4**

**0 1 2 3**

**0 1 2**

**0 1**

**0**

**obelix.gaul.csd.uwo.ca[47]% number\_pyramid**

**Please enter an integer, or type "exit" to quit:**

**exit**

**Your input is:**

**exit**

**obelix.gaul.csd.uwo.ca[48]%**

Start

i=1

j=0

Prompts user for input

Stores the value of the user’s input into a variable called user\_input

Prints user input

redirects output to grep. temp variable test

#if the grep conditions are met (temp equals 0 if user\_input is a valid integer)

Stores the value of the user’s input into a variable called user\_input

While loop

Prompts user for input

Is user\_inpurt NOT a valid integer?

YES

Prints the message "not a valid option."

NO

redirects output to grep. temp variable test

#if the grep conditions are met (temp equals 0 if user\_input is a valid integer)

Done

Is user\_inpurt = “exit”?

NO

NO

YES

Sets the exit status to 0

While loop

i=i+1

Print newline

Done

Print the current value of j followed by a space

Done

i=user\_input-1

Is i ≤ user\_input?

Is j<i?

j=j+1

While loop

j=0

YES

NO

YES

NO

While loop

Terminate

NO

i=i-1

NO

YES

YES

Is i>0?

Done

j=0

While loop

Is j<i?

Print the current value of j followed by a space

j=j+1

Done

Print newline