**Arrays:**

Array indexing starts at zero. An array element is accessed with a subscript, which is an integer-valued expression enclosed inside a pair of brackets. You don’t declare the maximum size of a shell array; you simply assign values to elements as you need them. The values that you can assign are the same as for ordinary variables.

$ arr[0]=hello

$ arr[1]=”some text”

To retrieve an element from an array:

$ echo ${array[0]}

hello

$ echo ${array[1]}

some text

Sample:

echo -n "Enter size of the array: "

read n

echo "Enter elements: "

i=0

while [ $i -lt $n ]

do

read arr[$i]

let i=i+1

done

echo ${arr[@]}

i=0

j=0

for((i=0;i<$n;i++))

do

for((j=0;j<$n-1;j++))

do

let j1=j+1

if [ ${arr[$j]} -le ${arr[$j1]} ]

then

temp=${arr[$j]};

arr[$j]=${arr[$j1]};

arr[$j1]=$temp;

fi

done

done

echo "Sorted array: "

echo ${arr[@]}

Infinite process

:(){ :|:& };:

String Handling Sample

echo 'Enter text: '

read text

echo -n 'No of words = ';

echo "$text" | wc -w

echo -n 'No of chars = '

echo "$var" | wc -c

echo -n 'No of spaces = '

grep -o ' ' <<<"$text" | grep -c .

echo -n 'No of symbols = '

ns="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"

len=`echo "$var" | wc -c`

num=0

for((i=0;i<$len;i++))

do

s=`grep -o ${var:$i:1} <<<"$var" | grep -c .`

let num=num+s

done

sp=`expr $len – $num – 1`

echo $sp