4/18/2023

Submitted to – Dr. Divya Gupta

Submitted by - Nisha

**Computer system architecture practical assignment**

**Name - nisha**

**Roll no.- 21pcs5123**

**Course - pscs**

**Semester - iv**

Q-1 Write a program to convert an unsigned number in one radix ‘A’ to the equivalent number in another radix ‘B’, where A and B can be any positive integer.

Input-

def val(c):

if (c >= '0' and c <= '9'):

return ord(c) -48

else:

return ord(c) -65 + 10

def toDeci(strr, base):

lenn = len(strr)

power = 1

num = 0

for i in range(lenn -1, -1, -1):

if (val(strr[i]) >= base):

print ("invalid number")

return -1

num += val(strr[i]) \*power

power = power \* base

return num

def reval(num):

if (num >= 0 and num <= 9):

return chr(num + 48)

else:

return chr(num -10 + 65)

def fromDeci (base, inputnum):

res = " "

while (inputnum > 0):

res += reval(inputnum % base)

inputnum //= base

res = res[:: -1]

return res

def convertBase (s, a, b):

num = toDeci(s,a)

ans = fromDeci (b,num)

print (ans)

def main():

s=str(input("enter a number:"))

a=int(input("enter a base of original number:"))

b=int(input("enter a base that you want a number:"))

convertBase(s,a,b)

if \_\_name\_\_=='\_\_main\_\_':

main()

Output-

enter a number:1100

enter a base of original number:2

enter a base that you want a number:10

12

Q-2 Write a program that will prompt for the input of two integer values. Then using the Bitwise shift operators show the result of

1. Left shifting the first number by the second
2. Right shifting the first number by the second

Input-

1. Left shifting the first number by the second

a=int(input("enter a number:"))

b=int(input("enter a number:"))

res=a<<b

print("Decimal is" , res)

def DecimaltoBinary(res):

if res>=1:

DecimaltoBinary(res//2)

print(res % 2 ,end='')

def main():

return DecimaltoBinary(res)

if \_\_name\_\_=="\_\_main\_\_":

main()

Output-

enter a number:1010

enter a number:4

Decimal is 16160

11111100100000

1. Right shifting the first number by the second

a=int(input("enter a number:"))

b=int(input("enter a number:"))

res=a>>b

print("Decimal is", res)

def DecimaltoBinary(res):

if res>=1:

DecimaltoBinary(res//2)

print(res % 2 ,end='')

def main():

return DecimaltoBinary(res)

if \_\_name\_\_=="\_\_main\_\_":

main()

Output-

enter a number:1011

enter a number:4

Decimal is 63

111111

Q-3 Write a program that will prompt for the input of two integer values. Then using the logical operators show the result of

1. Exclusive OR of the first number by the second bitwise
2. OR of the first number by the second bitwise
3. AND of the first number by the second bitwise

Input-

def DecimaltoBinary(res):

if res>=1:

DecimaltoBinary(res//2)

res1=res % 2

print(res1, end='')

def main():

a=int(input("enter a number:"))

b=int(input("enter a number:"))

choice=str(input("enter your choice(X:XOR/O:OR/A:AND)"))

if choice=="XOR" or choice=="X":

res=a^b

print("Decimal is:", res)

elif choice=="OR" or choice=="O":

res=a|b

print("Decimal is:", res)

elif choice=="AND" or choice=="A":

res=a&b

print("Decimal is:", res)

return DecimaltoBinary(res)

if \_\_name\_\_=='\_\_main\_\_':

main()

Output-a.

enter a number:15

enter a number:13

enter your choice(X:XOR/O:OR/A:AND)X

Decimal is: 2

10

Output-b.

enter a number:15

enter a number:13

enter your choice(X:XOR/O:OR/A:AND)O

Decimal is: 15

1111

Output-c.

enter a number:13

enter a number:15

enter your choice(X:XOR/O:OR/A:AND)A

Decimal is: 13

1101

Q-4 Write a program that will prompt for the input of a binary value and print

1. One’s complement
2. Two’s complement

Input-

def temp(c):

return '1' if (c=='0') else '0'

def printOnesAndTwosComplement (bin):

n=len(bin)

ones=" "

twos=" "

for i in range(n):

ones += temp(bin[i])

ones=list(ones.strip(""))

twos=list(ones)

for i in range (n-1, -1, -1):

if (ones[i]=='1'):

twos[i]='0'

else:

twos[i]='1'

break

print ("1's complement:", \*ones, sep="")

print ("2's complement:", \*twos, sep="")

def main():

print ("attention !!!")

print ("enter only binary number in the form of 0 and 1")

bin=str(input("enter a binary number:"))

printOnesAndTwosComplement (bin.strip(""))

if \_\_name\_\_=='\_\_main\_\_':

main()

Output-

attention !!!

enter only binary number in the form of 0 and 1

enter a binary number:11000

1's complement: 00111

2's complement: 01001

Q-6 Write a program to implement bit-wise addition of two numbers.

Input-

def max\_len(a,b):

max\_len=max(len(a), len(b))

a=a.zfill(max\_len)

b=b.zfill(max\_len)

result=' '

carry=0

for i in range (max\_len-1, -1, -1):

r=carry

r+=1 if a[i]=='1' else 0

r+=1 if b[i]=='1' else 0

result=('1' if r%2==1 else '0') + result

carry=0 if r<2 else 1

if carry != 0:

result ='1' + result

print(result.zfill (max\_len))

else:

print(result.zfill(max\_len))

def main():

a=str(input("enter a number:"))

b=str(input("enter a number:"))

return max\_len(a,b)

if \_\_name\_\_=='\_\_main\_\_':

main()

Output-

enter a number:1010

enter a number:1100

10110

Q-7 Write a program to implement bit-wise subtraction of two numbers using 2’s complement.

Input-

def subtract(n,a,b):

for i in range(n):

if (b[i] == 1):

b[i] = 0

else:

b[i] = 1

for i in range(n-1, -1, -1):

if (b[i] == 0):

b[i] = 1

break

else:

b[i] = 0

carry = 0

for i in range(n-1,-1,-1):

a[i] = a[i] + b[i] + carry

if (a[i] == 2):

a[i] = 0

carry = 1

elif (a[i] == 3):

a[i] = 1

carry = 1

else:

carry = 0

print()

if (carry == 1):

for i in range(n):

print(a[i],end="")

else:

for i in range(n):

if (a[i] == 1):

a[i] = 0

else:

a[i] = 1

for i in range(n-1, -1, -1):

if (a[i] == 0):

a[i] = 1

break

else:

a[i] = 0

print("-", end="")

for i in range(n):

print(a[i],end="")

def main():

#print("WARNINIG!!! ATTENTION PLEASE")

#print("enter only binary numbers")

#print("if you type any number rather than binary numbers it shows result")

s1=input("enter first binary number:")

s2=input("enter second binary number:")

maxlen=max(len(s1), len(s2))

s1=s1.zfill(maxlen)

s2=s2.zfill(maxlen)

l1=list(s1.strip(" "))

l2=list(s2.strip(" "))

s1=[int(i) for i in l1]

s2=[int(i) for i in l2]

subtract (maxlen,s1 ,s2)

if \_\_name\_\_=='\_\_main\_\_':

main()

Output-

enter first binary number:1101

enter second binary number:1010

0011

Q-8 Write a program to implement Selective Set logical operation using bit-wise operators.

Input-

def selective(a,b):

max\_len=max(len(a), len(b))

print("the length of bigger no.is:", max\_len)

a=a.zfill(max\_len)

b=b.zfill(max\_len)

print("a normalized", a)

print("b normalized", b)

#checking the number are valid or not

for i in range(0, max\_len):

if((a[i]!='0' and a[i]!='1') or (b[i]!='0' and b[i]!='1')):

print(a,b)

print("this is not a binary number")

print("enter binary numbers in the format of 0 or 1.\nfor cont")

cont = input("continue?/n:")

while cont=="y":

if \_\_name\_\_=='\_\_main\_\_':

main()

break

if cont=="n":

print("you choose n so the program has exit")

exit()

result=[0]\*max\_len

for i in range(0, len(result)):

if (a[i]=='0'):

if (b[i]=='0'):

result[i]='0'

else: result[i]='1'

else:

result[i]='1'

print("the selective set a will be:",' '.join(str(x) for x in result))

def main():

a=str(input("enter a number:"))

b=str(input("enter another number:"))

return selective(a,b)

if \_\_name\_\_=="\_\_main\_\_":

main()

Output-

enter a number:101010

enter another number:100011

the length of bigger no.is: 6

a normalized 101010

b normalized 100011

the selective set a will be: 1 0 1 0 1 1

Q-9 Write a program to implement Selective Complement logical operation using bit-wise operators.

Input-

def selective(a,b):

max\_len = max(len(a), len(b))

print("the length of bigger number is:", max\_len)

a=a.zfill(max\_len)

b=b.zfill(max\_len)

print("A normalized",a)

print("B normalized",b)

#checking the number are valid or not

for i in range(max\_len):

if ((a[i]!='0' and a[i]!='1') or (b[i] != '0' and b[i] != '1')):

print(a,b)

print("this is not a binary number")

print("enter binary number in the format of 0 or 1.\n for cont")

cont=input("continue?y/n:")

while cont.lower() == "y":

a=input("enter binary number A:")

b=input("enter binary number B:")

selective(a,b)

break

else:

print("you choose \"N\" so the program has exited!")

exit()

result=[0]\*max\_len

for i in range(0,len(result)):

if (a[i]=='0'):

if (b[i]=='0'):

result[i]='0'

else:

result[i]='1'

else:

result[i] = a[i]

selective\_comp = "".join(result)

print("the selective complement of a will be:", selective\_comp)

return selective\_comp

if \_\_name\_\_=='\_\_main\_\_':

a=input("enter binary number A:")

b=input("enter binary number B:")

selective(a,b)

Output-

enter binary number A:110

enter binary number B:101

the length of bigger number is: 3

A normalized 110

B normalized 101

the selective complement of a will be: 111

Q-10 Write a program to implement selective clear logical operation using bit-wise operators.

Input-

def selective(a,b):

max\_len = max(len(a), len(b))

print("the length of bigger number is:", max\_len)

a=a.zfill(max\_len)

b=b.zfill(max\_len)

print("a normalized",a)

print("b normalized", b)

#checking the number are valid or not

for i in range (0, max\_len):

if ((a[i]!='0' and a[i]!='1') or (b[i]!='0' and b[i]!='1')):

print(a,b)

print("this is not a binary number")

print("enter binary numbers in the format of 0 or 1.\n for cont")

cont = input("continue?y/n:")

while cont == 'y':

if \_\_name\_\_=='\_\_main\_\_':

main()

break

if cont == "n":

print("you choose n so the program has exit")

exit()

#getting the selective clear of a

result=[0]\*max\_len

for i in range(0, len(result)):

if (a[i]=='1'):

if (b[i]=='0'):

result[i]='1'

else:

result[i]='0'

else:

if(b[i]=='1'):

result[i]='0'

else:

result[i]='0'

print("the selective clear of a is:",' '.join(str(x) for x in result))

def main():

a=input('enter a number:')

b=input('enter another number:')

return selective(a,b)

if \_\_name\_\_=="\_\_main\_\_":

main()

Output-

enter a number:11100

enter another number:10011

the length of bigger number is: 5

a normalized 11100

b normalized 10011

the selective clear of a is: 0 1 1 0 0

THANK YOU