Computer science's Exam Solution

Exercice 1 (10 pts)

Trace the execution of the following C++ programs **(2×5 pts)**:

C++ Program	Trace
```cpp #include<iostream> using namespace std; int main (){ int a = 7,b=3; a+=b; b*=a; cout << "a"<<b; return 0;} ```	0.5 **a** | ~~7~~ 10 0.5 **b** | ~~3~~ 30 Screen: a 30 1
```cpp #include<iostream> using namespace std; int main (){ int a=15,b=16,c=-2; if(a<b){ b= a%7; c--; } cout<<"a"<<b<<"c"<<endl; return 0;} ```	0.25 **a** | 15 0.5 **b** | ~~16~~ 1 0.5 **c** | ~~-2~~ -3 Screen: a 1 c 0.75
```cpp #include<iostream> using namespace std; int main (){ for(int i=1;i>=6;i+=2){ cout<<i; } return 0;} ```	1 **i** | 1 Screen: 1
```cpp #include<iostream> using namespace std; int main (){ for(int i=1;i<10;i++){ if(i%2==0){ cout<<"Bonjour"<<endl; } } return 0;} ```	**i** | ~~1 2 3 4 5 6 7 8 9~~ 10 1 Screen: Bonjour / Bonjour / Bonjour 1 / Bonjour
```cpp #include<iostream> using namespace std; int main(){ int n=-1,x=5; x-=4; n=n+x; if(n>0){ cout<<"Positive number"<<endl; } else if(n<0){ cout<<"Negative number"<<endl; }else{ cout<<"zeros number"; } return 0;} ```	0.5 **n** | ~~-1~~ 0 0.5 **x** | ~~5~~ 1 Screen: zeros number 1

Exercise 2 (6 pts)

Do the following base conversions (show details)

From binary to decimal: 11011	From decimal to binary: 37
$11011_2 = 1*2^0 + 1*2^1 + 0*2^2 + 1*2^3 + 1*2^4$ $\quad = 1+2+8+16$ $\quad = 27$ 1	$37/2=18$ remainder 1 $18/2=9$ remainder 0 $9/2=4$ remainder 1 $37_{10}= 100101_2$ $4/2=2$ remainder 0 $2/2=1$ remainder 0 1 $1/2=0$ remainder 1
From binary to octal: 1101101	From octal to binary: 123
$1101101_2= (001)(101)(101)$ $\quad = 155_8$ 1	$123_8 = (001)(010)(011)$ $\quad = 1010011_2$ 1
From octal to decimal: 129	From decimal to octal: 87
This number is not a valid octal number 1	$87 /8 =10$ remainder 7 $10/8= 1$ remainder 2 $1/8=0$ remainder 1 1 $87_{10}=127_8$

Exercise 3 (4 pts)

Given a positive integer n by the user, write a C++ program to check if this number is perfect or not. A number is said to be perfect if sum of all its factors excluding the number itself is equal to this number. Examples:

- N = 6. Factors of 6 are 1, 2, 3 and 6. Excluding 6 their sum is 6 which is equal to N itself. So, 6 is a Perfect Number.
- N = 10. Factors of 10 are 1, 2, 5 and 10. Excluding 10 their sum is 8 which is not equal to N itself. So, 10 is not a Perfect Number.

```cpp
#include<iostream>
using namespace std;
int main (){
int n,s=0;                                    0.5
cout<<"Please enter a positive integer ";     0.25
cin>>n;                                        0.25
for(int i=1;i<=n/2;i++){                       0.5
if(n%i==0){                                     0.5
s=s+i;                                          0.5
}
}
if(s==n){                                       0.5
cout<<n<<" is perfect"<<endl;                  0.25
}else{                                          0.5
cout<<n<<" is not perfect"<<endl;              0.25
}
return 0;
}
```