|  |  |  |
| --- | --- | --- |
|  | **Computers Engineering @ AI Dept.**  Data Structures (Recursion) |  |

**Sheet 01**

Assume any missing data.

1. Write a C++ program to find sum of digits of a number using recursion.
2. In problem 1, write the steps of recursion program that you have followed.
3. Write a C++ program to find product of 2 numbers using recursion. (this program do multiplication as a repeated sums, i.e., 12\*3= 12+12+12)
4. Write an algorithm and a C++ program to convert binary code of a number (e.g., 4-bit) into its equivalent Gray code using recursion.

Hints: The Gray code is an ordering of the binary system such that two successive values differ in only one bit. Gray codes are very useful in the normal sequence of binary numbers generated by the hardware that may cause an error or ambiguity during the transition from one number to the next. So, the Gray code can eliminate this problem easily since only one bit changes its value during any transition between two numbers.

|  |  |  |
| --- | --- | --- |
| **Decimal** | **Binary** | **Gray** |
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |
| 9 | 1001 | 1101 |
| 10 | 1010 | 1111 |
| 11 | 1011 | 1110 |
| 12 | 1100 | 1010 |
| 13 | 1101 | 1011 |
| 14 | 1110 | 1001 |
| 15 | 1111 | 1000 |

// A C++ Program to find Sum of Digits of a Number using Recursion

#include <iostream>

int sum (int a);

int main()

{

int num, result;

std::cout<<"Enter the number: ";

std::cin>> num;

result = sum(num);

std::cout<<"Sum of digits in " << num <<" is " <<result;

return 0;

}

int sum (int num)

{

if (num != 0)

{

return (num % 10 + sum (num / 10));

}

else

{

return 0;

}

}

// A C++ Program to find Product of 2 Numbers using Recursion

#include <iostream>

int product(int, int);

int main()

{

int a, b, result;

std::cout<<"Enter two numbers to find their product: ";

std::cin>> a;

std::cin>> b;

result = product(a, b);

std::cout<<"Product of " << a <<" and " << b << " is "<< result;

return 0;

}

int product(int a, int b)

{

if (a < b)

{

return product(b, a);

}

else if (b != 0)

{

return (a + product(a, b - 1));

}

else

{

return 0;

}

}

// A C++ Program to Convert Binary Code of a Number into its Equivalent Gray Code using Recursion

#include <iostream>

int bin\_gray(int);

int main ()

{

int bin, gray;

std::cout<<"Enter a binary number: ";

std::cin>> bin;

gray = bin\_gray (bin);

std::cout<<"The gray code of " << bin<< " is " <<gray;

return 0;

}

int bin\_gray (int bin)

{

int a, b, result = 0, i = 0;

if (!bin)

{

return 0;

}

else

{

a = bin % 10;

bin = bin / 10;

b = bin % 10;

if ((a && !b) || (!a && b))

{

return (1 + 10 \* bin\_gray (bin));

}

else

{

return (10 \* bin\_gray (bin));

}

}

}