/\*

KAHLIL BELLO

DATA STRUTURES

SEPTEMBER 19, 2018

project 3 prob 1

\*/

#include <iostream>

using namespace std;

class STACK

{

private:

int a[16];

int counter;

public:

void clearStack()

{

counter = 0;

}//end clearStack

bool emptyStack()

{

return (counter == 0) ? true : false;

}//end emptyStack

bool fullStack()

{

return (counter == 16) ? true : false;

}//end fullStack

void pushStack(int x)

{

a[counter] = x;

counter++;

}//end pushStack

int popStack()

{

counter--;

return a[counter];

}

};//end class STACK

int main()

{

STACK base2;

int n,n2,n3, r;

base2.clearStack();

cout << "Enter an integer to be converted to binary, octal, and hexadecimal. . .\n";

cin >> n;

n2 = n;

n3 = n;

cout << n << " at base 2 is ";

while (n != 0)

{

r = (n % 2);

base2.pushStack(r);

n = (n / 2);

}

while (!base2.emptyStack())

{

int x = base2.popStack();

cout << x;

}

cout << endl;

//###############OCTAL##########################

r = 0;

STACK base8;

base8.clearStack();

cout << n2 << " at base 8 is ";

while (n2 != 0)

{

r = (n2 % 8);

base8.pushStack(r);

n2 = (n2 / 8);

}

while (!base8.emptyStack())

{

int y = base8.popStack();

cout << y;

}

cout << endl;

//###############HEXADECIMAL##########################

STACK base16;

r = 0;

base16.clearStack();

cout << n3 << " at base 16 is ";

while (n3 != 0)

{

r = (n3 % 16);

base16.pushStack(r);

n3 = (n3 / 16);

}

while (!base16.emptyStack())

{

int x = base16.popStack();

if(x<=9)

{

cout << x;

}

else

{

switch (x)

{

case 10: cout << 'A';

break;

case 11: cout << 'B';

break;

case 12: cout << 'C';

break;

case 13: cout << 'D';

break;

case 14: cout << 'E';

break;

case 15: cout << 'F';

break;

}//end switch

}

}

cout << "\n\n";

system("pause");

return 0;

}

/\*

KAHLIL BELLO

DATA STRUTURES

SEPTEMBER 19, 2018

project 3 prob 1

OUTPUT:

Enter an integer to be converted to binary, octal, and hexadecimal. . .

163

163 at base 2 is 10100011

163 at base 8 is 243

163 at base 16 is A3

Press any key to continue . . .

\*/

