stack.h

```
#ifndef STACK_H_INCLUDED

#define STACK_H_INCLUDED

#include <string>

#include <iostream>


using namespace std;


typedef int infotype;


struct Stack {

   int info[20];

   int top;

};


void createStack(Stack &S);

void push(Stack &S, infotype X);

infotype pop (Stack &S);

void printInfo(Stack &S);

void balikStack(Stack &S);

void pushAscending(Stack &S, int X);



#endif // STACK_H_INCLUDED
```

stack.cpp

```
#include "Stack.h"

void createStack(Stack &S){
    S.top = -1;
}

void push(Stack &S, infotype X){
    if (S.top == -1){
        S.top = 0;
        S.info[0] = X;
    } else {
        S.top = S.top + 1;
        S.info[S.top] = X;
    }
}
infotype pop (Stack &S){
    infotype X;
    if (S.top == 0){
        X = S.info[0];
        S.top = -1;
        return X;
    } else {
        if (S.top != -1){
            X = S.info[S.top];
            S.top = S.top - 1;
            return X;
        }
    }
}
```

```
void printInfo(Stack &S){
   if (S.top != -1){
      cout << "[TOP] ";
      int i;
      for (i=S.top;i>=0;i--){
         cout << S.info[i] << " ";
      }
   } else {
      cout << "Stack kosong" << endl;
   }
}
void balikStack(Stack &S){
   Stack S1;
   S1 = S;
   createStack(S);
   int X;
   while(S1.top != -1){
      X = pop(S1);
      push(S, X);
   }
}

void pushAscending(Stack &S, int X){
   if(S.top == -1){
      S.top = 0;
      S.info[S.top] = X;
   } else if(S.top == 0){
      S.top = S.top + 1;
      if(S.info[S.top-1] > X)
      {
            int temp = S.info[S.top-1];
            S.info[S.top-1] = X;
            S.info[S.top] = temp;
        }
        else if(S.info[S.top-1] < X)
        {
            S.info[S.top] = X;
        }
    } else
    {
        S.top = S.top + 1;
        S.info[S.top] = X;

        for(int i = 0; i < S.top - 1; i++)
        {
            int min_idX = S.info[i];
            int loc = i;
            for(int j = i + 1; j < S.top; j++)
            {
                if(min_idX > S.info[j])
                {
                    min_idX = S.info[j];
                    loc = j;
                }
            }
            int tmp = S.info[i];
            S.info[i] = S.info[loc];
            S.info[loc] = tmp;
        }
    }
}
```

main.cpp

```
#include "Stack.h"
int main()
{
    cout << "Hello World!" << endl;

    int X;
    Stack S, S1;
    createStack(S);

    push(S, 3);
    push(S, 4);
    push(S, 8);
    X = pop(S);
    push(S, 2);
    push(S, 3);
    X = pop(S);
    push(S, 9);
    printInfo(S);
    cout << endl;
    cout << "balik stack" << endl;
    balikStack(S);
    printInfo(S);

    cout << endl << endl;
    cout << "Hello World!" << endl;

    createStack(S1);

    pushAscending(S1, 3);
    pushAscending(S1, 4);
    pushAscending(S1, 8);
    pushAscending(S1, 2);
    pushAscending(S1, 3);
    pushAscending(S1, 9);
    printInfo(S1);
    cout << endl;
    cout << "balik stack" << endl;
    balikStack(S1);
    printInfo(S1);


    return 0;
}
```