//COS10007 – Developing Technical Software//

//Lab 2 – Son Nguyen

//Student ID : 103234103//

1/

//COS10007 – Developing Technical Software//

//Lab 2 – Son Nguyen

//Student ID : 103234103//

//(a)

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct bankEmployee {

char name[20];

int salary;

struct bankEmployee \*next;

};

typedef struct bankEmployee BANKEmployee;

typedef BANKEmployee \*BANKEmployeePtr;

int main()

{

//declaring pointer

BANKEmployeePtr startPtr = NULL;

BANKEmployeePtr newPtr;

BANKEmployeePtr currentPtr;

BANKEmployeePtr previousPtr;

BANKEmployeePtr tempPtr;

//(b) Justin

newPtr=(BANKEmployee\*)malloc(sizeof(BANKEmployee));

strcpy(newPtr->name, "Justin");

newPtr-> salary = 1000;

newPtr-> next="Sam";

startPtr=newPtr;

//(c)

// 1C - Sam Section

newPtr=(BANKEmployee\*)malloc(sizeof(BANKEmployee));

strcpy(newPtr -> name, "Sam");

newPtr -> salary = 999;

newPtr -> next= NULL;

startPtr->next= newPtr;

// 1C - Antony

newPtr=(BANKEmployee\*)malloc(sizeof(BANKEmployee));

strcpy(newPtr -> name, "Antony");

newPtr -> salary = 200;

currentPtr= startPtr; //swpping startPTR with currentPTR

newPtr -> next = currentPtr;

startPtr = newPtr;

//1C - Tony

newPtr=(BANKEmployee\*)malloc(sizeof(BANKEmployee));

strcpy(newPtr -> name, "Tony");

newPtr -> salary = 300;

newPtr -> next = NULL;

previousPtr = currentPtr -> next;

previousPtr -> next = newPtr;

//1C - Peter

newPtr=(BANKEmployee\*)malloc(sizeof(BANKEmployee));

strcpy(newPtr -> name, "Peter");

newPtr -> salary = 400;

//making connection b/w Peter and Justin

currentPtr -> next = newPtr;

//making connection b/w Peter and Justin

newPtr -> next = previousPtr;

//(d)

while(currentPtr!=NULL)

{

printf("Name: %s \n Salary: %d \n" , currentPtr -> name, currentPtr -> salary);

currentPtr = currentPtr->next;

}

//(e)

currentPtr = startPtr;

while (currentPtr != NULL)

{

tempPtr = currentPtr;

currentPtr = currentPtr-> next;

free(tempPtr);

}

return (0);

}

**Example Output**   
Text

Description automatically generated  
  
2/  
#include <stdio.h>

#include <stdlib.h>

//34103

struct myID {

int value;

struct myID \*next;

};

typedef struct myID MYID;

typedef MYID \* MYIDPtr;

int main() {

MYIDPtr head = NULL;

MYIDPtr newptr, current;

// create first digit

newptr = (MYIDPtr) malloc(sizeof(MYID));

newptr->value = 3;

newptr->next = NULL;

head = newptr;

// create second node

newptr = (MYIDPtr) malloc(sizeof(MYID));

newptr->value = 4;

newptr->next = NULL;

current = head;

while (current->next != NULL && current->next->value < newptr->value) {

current = current->next;

}

newptr->next = current->next;

current->next = newptr;

// create third node

newptr = (MYIDPtr) malloc(sizeof(MYID));

newptr->value = 1;

newptr->next = NULL;

current = head;

while (current->next != NULL && current->next->value < newptr->value) {

current = current->next;

}

newptr->next = current->next;

current->next = newptr;

// create fourth node

newptr = (MYIDPtr) malloc(sizeof(MYID));

newptr->value = 0;

newptr->next = NULL;

current = head;

while (current->next != NULL && current->next->value < newptr->value) {

current = current->next;

}

newptr->next = current->next;

current->next = newptr;

// create fifth node

newptr = (MYIDPtr) malloc(sizeof(MYID));

newptr->value = 3;

newptr->next = NULL;

current = head;

while (current->next != NULL && current->next->value < newptr->value) {

current = current->next;

}

newptr->next = current->next;

current->next = newptr;

// print the linked list

current = head;

while (current != NULL) {

printf("%d ", current->value);

current = current->next;

}

printf("\n");

return 0;

}  
Example Output:

