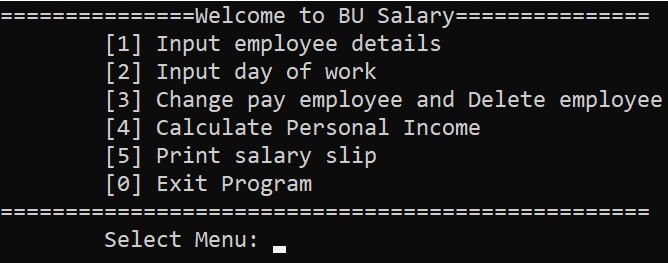
**การทำงานของโปรแกรม**



หน้านี้จะเป็นหน้าเมนูหลักเมื่อเริ่มเปิดโปรแกรมโดยจะมีเมนูให้เลือกอยู่ถึง 5 เมนูโดยให้พิมพ์ตัวเลขของเมนูที่ต้องการจะใช้งานโดยเลข 0 จะเป็นการออกจากโปรแกรม

โดยที่เมนูของโปรแกรมนี้มีทั้งหมด 6 เมนู ประกอบไปด้วย

1. **การใส่ค่ารายละเอียดของพนักงาน**

ใช้โครงสร้างการเก็บข้อมูลแบบ BST อ่านค่าจากไฟล์ txt และการเก็บข้อมูลผ่านทางแป้นพิมพ์

1. **การใส่วันเวลาในการทำงานของพนักงาน**

แทรกข้อมูลเวลาในการทำงานในโดยใช้โครงสร้างแบบ BST

1. **เปลี่ยนรายละเอียดการจ่ายเงินของพนักงานและการลบข้อมูลพนักงาน**

เปลี่ยนรายละเอียดข้อมูลในโครงสร้าง BST และการลบ node

1. **การคำนวณรายได้ทั้งหมดของพนักงาน**

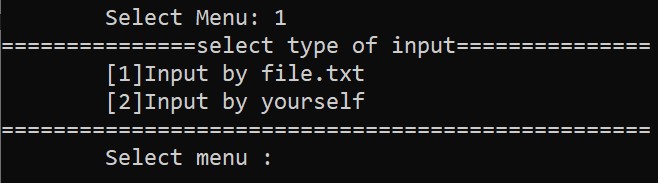
คำนวณค่ารายได้ของพนักงานที่ได้จากข้อมูลทั้งหมดใน BST

1. **การปริ้นข้อมูลของพนักงาน**

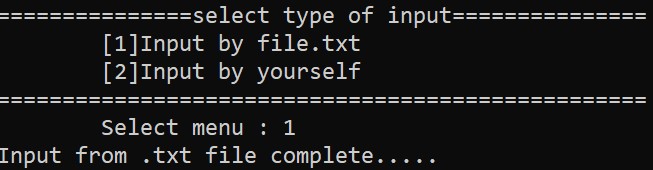
แสดงข้อมูลของพนักงาน รายได้ที่ได้และอื่นๆ ผ่านหน้าจอ และ textfile

1. **ออกจากโปรแกรม**

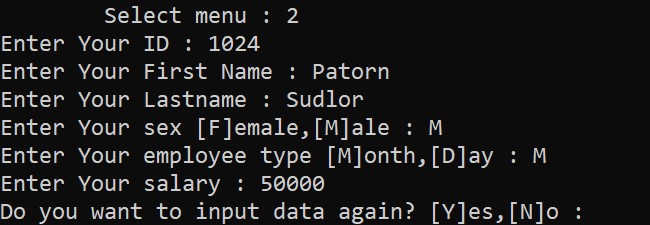
**เมนูที่ 1 Input employee detail**



`เมื่อพิมพ์เลข 1 ก็จะเข้ามาที่เมนู Input employee details โดยเมนูนี้จะทำหน้าที่เพิ่มรายละเอียดของพนักงานโดยสามารถเลือกได้ว่าจะรีบค่าจาก .txt ไฟล์ หรือรับค่าผ่านทางแป้นพิมม์

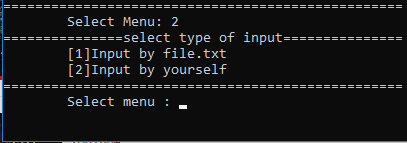


เมื่อพิมพ์เลข 1 โปรแกรมก็จะรับค่าจากไฟล์ employee.txt แล้วจะกลับไปยังหน้าเมนูหลัก

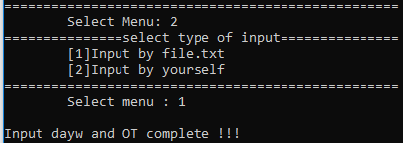


เมื่อพิมพ์เลข 2 โปรแกรมจะเป็นการรับค่ารายละเอียดพนักงานผ่านทางคีย์บอร์ดแล้วโปรแกรมจะถามว่าต้องการจะเพิ่มรายละเอียดพนักงานต่อไหมถ้าต้องการให้พิมพ์ Y ก็จะให้เพิ่มรายละเอียดพนักงานต่อแต่ถ้าพิมพ์ N ก็จะกลับไปหน้าเมนูหลัก

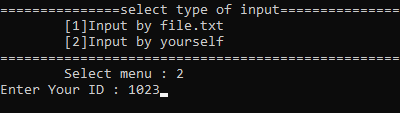
**เมนูที่ 2 Input day of work**



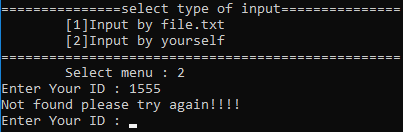
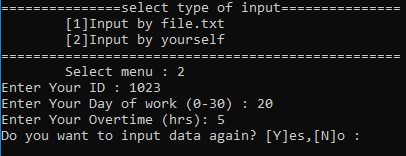
เมื่อพิมพ์เลข 2 ในเมนูหลักจะพบกับเมนูให้เลือกตามรูปข้างต้นอีก 2 เมนู คือการใส่ค่าของวันที่ทำงาน และ OT ผ่านทาง file.txt หรือการ ใส่ค่าผ่านทางแป้นพิมพ์ด้วยตนเอง



หลังจากเข้าเมนูรองมาแล้วเมื่อเลือกหัวข้อที่ 1 คือ ใส่ค่าผ่านทาง file.txt โปรแกรมก็จะทำการใส่ค่า วันทำงานกับ OT เข้ามาและจะแสดงผลบอกว่าใส่ค่าเสร็จเรียบร้อยแล้ว ตามภาพข้างต้น

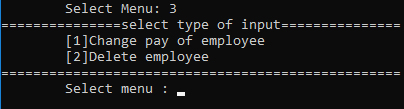


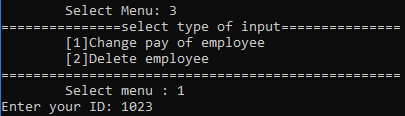
เมื่อเลือกหัวข้อที่ 2 จะเป็นการใส่ข้อมูลเวลาด้วยตนเอง เมื่อเลือกหัวข้อนี้เข้ามาโปรแกรมจะให้ใส่ ID ของคนที่ต้องการจะใส่เวลาทำงานและ OT



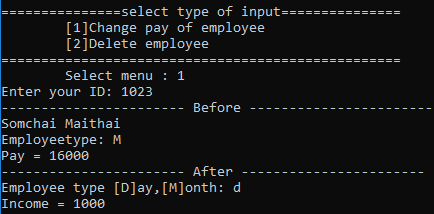
เมื่อ ID ที่กรอกมีข้อมูลเพื่อที่จะให้สำหรับใส่ วันที่และเวลาเพิ่มก็จะให้ใส่ แต่ถ้ากรอก ID ที่ไม่มีอยู่ในฐานข้อมูลโปรแกรมจะแจ้ง ตามรูปข้างต้นในตอนท้ายโปรแกรมจะถามว่าต้องการใส่เพิ่มไหมถ้าใช่ตอบ Y ถ้าไม่ใช่ตอบ N

**เมนูที่ 3 Change pay employee and Delete employee**

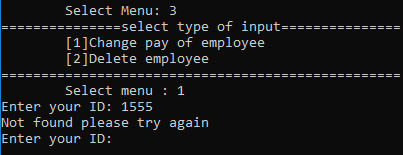


เมื่อเลือกเมนูที่ 3 จากเมนูหลักจะเข้ามาที่เมนูรองของหัวข้อนี้คือให้เลือก ว่าจะทำการเปลี่ยนข้อมูลของพนักงานหรือจะทำการลบข้อมูลพนักงาน

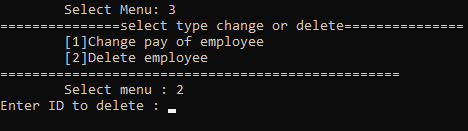
เมื่อเลือกเมนูที่ 1 ในเมนูรองจะพบว่าให้ใส่ ID ที่จะทำการเปลี่ยนข้อมูล



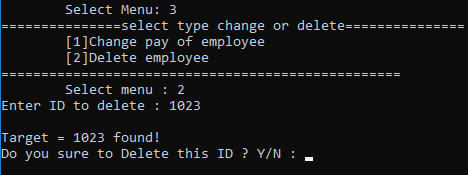
ถ้า ID ที่ใส่มีตามฐานข้อมูลของระบบโปรแกรมจะแจ้งข้อมูลก่อนการเปลี่ยนแปลงและให้ใส่ข้อมูลที่ต้องการที่จะเปลี่ยนแปลง



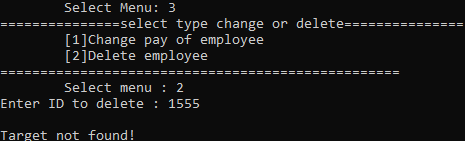
หาก ID ที่ใส่ไม่มีตามฐานข้อมูลของระบบโปรแกรมจะแจ้งว่า ไม่พบข้อมูลและให้กรอก ID ที่ต้องการจะเปลี่ยนแปลงใหม่



เมื่อเลือกเมนูที่ 2 โปรแกรมจะให้ใส่ ID ของพนักงานที่ต้องการจะลบ

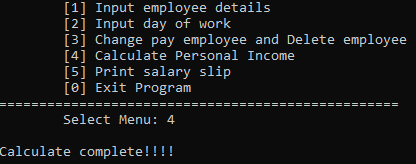


เมื่อกรอก ID ที่มีอยู่ในฐานข้อมูล โปรแกรมจะแจ้งว่าเจอข้อมูลนั้นและถามซ้ำอีกครั้งว่าต้องการจะลบหรือไม่เมื่อตอบ Y ก็จะลบ ข้อมูลนั้นทันที เมื่อตอบ N ก็จะออกจากหน้านี้



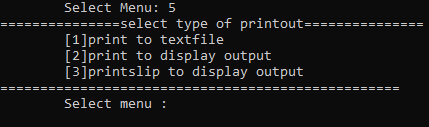
เมื่อกรอก ID ที่จะลบแต่ไม่มีในฐานข้อมูลโปรแกรมจะแจ้งว่าไม่พบ ID ที่ต้องการลบ และจะกลับไปที่เมนูหลัก

**เมนูที่ 4 Calculate Personal Income**

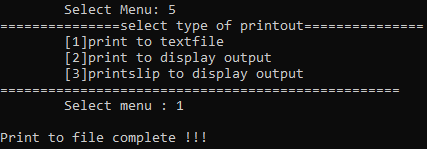


เมื่อเลือกเมนูที่ 4 จากเมนูหลัก โปรแกรมจะทำการคำนวณค่าทั้งหมดและเก็บข้อมูลลงใน BST และแจ้งว่า คำนวณเสร็จเรียบร้อยแล้ว

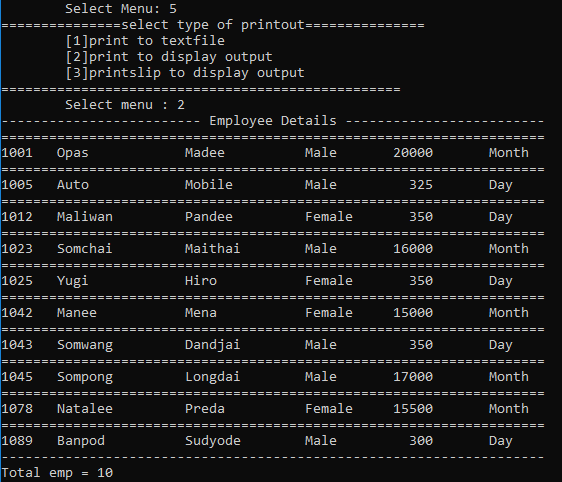
**เมนูที่ 5 Print salary slip**

****

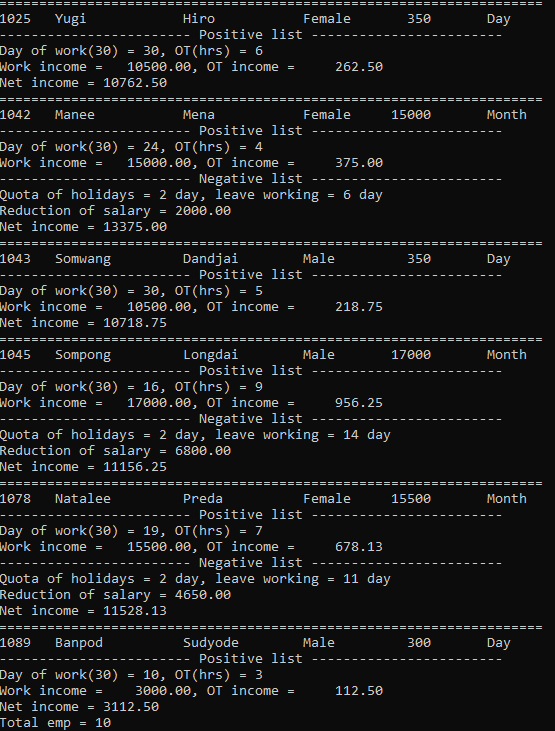
เมื่อเลือกเมนูที่ 5 จากเมนูหลักจะมี 3 เมนูรองให้เลือกคือ ปริ้นทาง text.file ปริ้นทางหน้าจอ และปริ้นเป็นสลิปรายละเอียดทั้งหมดผ่านทางหน้าจอ



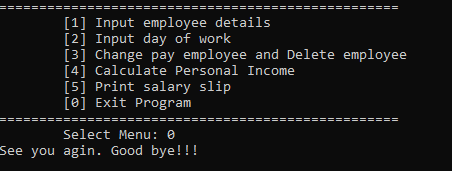
เมื่อเลือกเมนูที่ 1 จากเมนูรอง โปรแกรมจะทำการปริ้นข้อมูลทั้งหมดลง textfile และแจ้งผ่านทางหน้าจอว่า ปริ้นเข้าไฟล์เรียบร้อยแล้ว



เมื่อเลือกเมนูที่ 2 จากเมนูรอง โปรแกรมจะทำการปริ้นข้อมูลทั้งหมดผ่านทางหน้าจอจะแสดงผลตามรูปข้างต้น



เมื่อเลือกเมนูที่ 3 จากเมนูรอง โปรแกรมจะทำการแสดงสลิปและบอกว่า รายได้เท่าไหร่ หักเท่าไหร่ แล้วรายได้รวมทั้งหมดกี่บาท



และหลังจากนั้นก็คือ เมนู 0 คือการออกจากโปรแกรมนั้นเอง

**ภาคผนวค**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include <stdlib.h>

typedef struct{

int id,dumid,pay,dayw,ot,changeP,dayh,day,proday;

char fname[20],lname[20],sex,emptype,bin[100],changeT;

float net,tax,deduction,payot,netot,income; //payroll deduction

}empinfo;

typedef struct node{

empinfo data;

struct node \*right,\*left;

}bst;

//===================================================[1] INPUT FILE============================================================

int inputmenu()

{

int slinput=0;

printf("===============select type of input===============\n");

printf(" [1]Input by file.txt\n");

printf(" [2]Input by yourself\n");

printf("==================================================\n");

printf(" Select menu : ");

scanf("%d",&slinput);

return slinput;

}

void insertbst(bst \*\*root,empinfo newdata,int \*cnt)

{

bst \*newnode,\*run,\*prev;

newnode = malloc(sizeof(bst));

newnode->data = newdata;

newnode->left = NULL;

newnode->right = NULL;

if(\*root == NULL)

{

\*root = newnode;

}

else

{

run = \*root;

while(run != NULL)

{

prev = run;

if(newnode->data.id > run->data.id)

{

run = run->right;

}

else if(newnode->data.id < run->data.id)

{

run = run->left;

}

}

if(newnode->data.id > prev->data.id)

{

prev->right = newnode;

}

else if(newnode->data.id < prev->data.id)

{

prev->left = newnode;

}

}

(\*cnt)++;

}

void createbst(bst \*\*root,int \*cnt)

{

FILE \*inf;

inf = fopen("employee.txt","r");

empinfo newdata;

fgets(newdata.bin,100,inf);

newdata.day = 30;

newdata.proday = 2;

fscanf(inf,"%d%s%s %c%d %c",&newdata.id,&newdata.fname,&newdata.lname,&newdata.sex,&newdata.pay,&newdata.emptype);

while(!feof(inf))

{

insertbst(&\*root,newdata,cnt);

fscanf(inf,"%d%s%s %c%d %c",&newdata.id,&newdata.fname,&newdata.lname,&newdata.sex,&newdata.pay,&newdata.emptype);

}

fclose(inf);

}

int checkid(bst \*root,empinfo newdata)

{

bst \*run;

run = root;

if(run != NULL)

{

checkid(run->left,newdata);

if(newdata.id == run->data.id)

{

return 1;

}

checkid(run->right,newdata);

}

}

void inputkeyboard(bst \*\*root,int \*cnt)

{

int check=-1;

char ans='Y';

empinfo newdata;

do

{

printf("Enter Your ID : ");

scanf("%d",&newdata.id);

check = checkid(\*root,newdata);

if(check != 1)

{

printf("Enter Your First Name : ");

scanf("%s",&newdata.fname);

printf("Enter Your Lastname : ");

scanf("%s",&newdata.lname);

printf("Enter Your sex [F]emale,[M]ale : ");

scanf(" %c",&newdata.sex);

printf("Enter Your employee type [M]onth,[D]ay : ");

scanf(" %c",&newdata.emptype);

printf("Enter Your salary : ");

scanf("%d",&newdata.pay);

printf("Do you want to input data again? [Y]es,[N]o : ");

scanf(" %c",&ans);

newdata.day = 30;

newdata.proday = 2;

printf("==================================================\n");

insertbst(&\*root,newdata,cnt);

}

else

{

printf("duplicate ID !!!!\n");

}

}while(toupper(ans) == 'Y');

}

//=============================================================================================================================

//===================================================[2] INPUT DAY OF WORK=====================================================

int daywmenu()

{

int sldayw=0;

printf("===============select type of input===============\n");

printf(" [1]Input by file.txt\n");

printf(" [2]Input by yourself\n");

printf("==================================================\n");

printf(" Select menu : ");

scanf("%d",&sldayw);

return sldayw;

}

void insertdayw(bst \*root,empinfo newdata)

{

bst \*run;

run = root;

if(run != NULL)

{

insertdayw(run->left,newdata);

if(newdata.dumid == run->data.id)

{

run->data.dayw = newdata.dayw;

run->data.ot = newdata.ot;

}

insertdayw(run->right,newdata);

}

}

void filedayw(bst \*root)

{

FILE \*inf;

inf = fopen("work.txt","r");

empinfo newdata;

fgets(newdata.bin,100,inf);

fscanf(inf,"%d%d%d",&newdata.dumid,&newdata.dayw,&newdata.ot);

while(!feof(inf))

{

insertdayw(root,newdata);

fscanf(inf,"%d%d%d",&newdata.dumid,&newdata.dayw,&newdata.ot);

}

fclose(inf);

}

void inputdayw(bst \*root)

{

int check = 0;

char ans;

empinfo newdata;

do

{

printf("Enter Your ID : ");

scanf("%d",&newdata.dumid);

newdata.id = newdata.dumid;

check = checkid(root,newdata);

if(check == 1)

{

printf("Enter Your Day of work (0-30) : ");

scanf("%d",&newdata.dayw);

printf("Enter Your Overtime (hrs): ");

scanf("%d",&newdata.ot);

insertdayw(root,newdata);

printf("Do you want to input data again? [Y]es,[N]o : ");

scanf(" %c",&ans);

}

else

{

printf("Not found please try again!!!!\n");

inputdayw(root);

}

}while(toupper(ans) == 'Y');

}

//=============================================================================================================================

//===================================================[3] Change pay of employee=================================================================

int changemenu()

{

int slchange=0;

printf("===============select type change or delete===============\n");

printf(" [1]Change pay of employee\n");

printf(" [2]Delete employee\n");

printf("==================================================\n");

printf(" Select menu : ");

scanf("%d",&slchange);

return slchange;

}

void printpay(bst \*root,empinfo newdata)

{

bst \*run;

run = root;

if(run != NULL)

{

printpay(run->left,newdata);

if(newdata.dumid == run->data.id)

{

printf("----------------------- Before -----------------------\n");

printf("%s %s\n",run->data.fname,run->data.lname);

printf("Employeetype: %c\n",run->data.emptype);

printf("Pay = %d\n",run->data.pay);

}

printpay(run->right,newdata);

}

}

int insertpay(bst \*root,empinfo newdata)

{

bst \*run;

run = root;

if(run != NULL)

{

insertpay(run->left,newdata);

if(newdata.dumid == run->data.id)

{

run->data.emptype = newdata.changeT;

run->data.pay = newdata.changeP;

}

insertpay(run->right,newdata);

}

}

bst\* changeP(bst \*root)

{

int check = 0;

empinfo newdata;

printf("Enter your ID: ");

scanf("%d",&newdata.dumid);

newdata.id = newdata.dumid;

check = checkid(root,newdata);

if(check == 1)

{

printpay(root,newdata);

printf("----------------------- After -----------------------\n");

printf("Employee type [D]ay,[M]onth: ");

scanf(" %c",&newdata.changeT);

newdata.changeT=toupper(newdata.changeT);

if(toupper(newdata.changeT) == 'D' || toupper(newdata.changeT) == 'M')

{

printf("Income = ");

scanf("%d",&newdata.changeP);

insertpay(root,newdata);

return root;

}

else

printf("Please key D or M.\n");

root = changeP(root);

}

else

printf("Not found please try again\n");

root = changeP(root);

}

void deleteNode(bst \*\*root,bst \*prev,bst \*run)

{

bst \*temp;

//delete leaf node

if(run->left == NULL && run->right == NULL)

{

if(run == \*root)

\*root = NULL;

else

if(prev->left == run)

prev->left = NULL;

else

prev->right = NULL;

}

else if(run->left != NULL && run->right != NULL)

{

prev = run;

temp = run->left;

while(temp->right != NULL)

{

prev = temp;

temp = temp->right;

}

run->data = temp->data;

if(prev == run)

prev->left = temp->left;

else

prev->right = temp->left;

free(temp);

}

else if(run->left != NULL)

{

if(run == \*root)

\*root = run->left;

else

{

if(prev->left == run)

prev->left = run->left;

else

prev->right = run->left;

}

}

else

{

if(run == \*root)

\*root = run->right;

else

{

if(prev->left == run)

prev->left = run->right;

else

prev->right = run->right;

}

free(run);

}

}

void deleteBST(bst \*\*root,int \*cnt)

{

bst \*run,\*prev;

run = \*root;

int target=0;

char sure;

printf("Enter ID to delete : ");

scanf("%d",&target);

while(run != NULL && target != run->data.id)

{

prev = run;

if(target > run->data.id)

run = run->right;

else

run = run->left;

}

if(run == NULL)

{

printf("\nTarget not found!\n");

//return 0;

}

else

{

printf("\nTarget = %d found!\n",target);

printf("Do you sure to Delete this ID ? Y/N : ");

scanf(" %c",&sure);

if(toupper(sure) == 'Y')

{

(\*cnt)--;

deleteNode(&\*root,prev,run);

}

else if(toupper(sure) == 'N')

{

printf("goodbye !!!\n");

start();

}

}

}

//=============================================================================================================================

//===================================================[4] Calculate Personal Income Tax=================================================================

bst\* calculatepay(bst \*root)

{

bst \*run;

run = root;

if(run != NULL)

{

calculatepay(run->left);

run->data.dayh = run->data.day - run->data.dayw;

if(toupper(run->data.emptype) == 'D') //type D OT 1.0

{

run->data.income = run->data.dayw \* run->data.pay;

run->data.payot = (float)run->data.pay / 8; //calculate OT per hour

run->data.netot = run->data.ot \* run->data.payot;

run->data.net = run->data.income + run->data.netot;

}

else if(toupper(run->data.emptype) == 'M') //type M OT 1.5

{

run->data.income = run->data.pay;

run->data.payot = (((float)run->data.pay / run->data.day) / 8) \* 1.5; //calculate OT per hour

run->data.netot = run->data.ot \* run->data.payot;

if(run->data.dayh <= 2)

run->data.net = run->data.pay + run->data.netot;

else

{

run->data.deduction = ((float)run->data.pay/run->data.day)\*(run->data.dayh - run->data.proday);

run->data.net = ((float)run->data.pay + run->data.netot) - run->data.deduction;

}

}

calculatepay(run->right);

}

return root;

}

//=============================================================================================================================

//===================================================[5] PRINT=================================================================

int printmenu()

{

int slprint=0;

printf("===============select type of printout===============\n");

printf(" [1]print to textfile\n");

printf(" [2]print to display output\n");

printf(" [3]printslip to display output\n");

printf("==================================================\n");

printf(" Select menu : ");

scanf("%d",&slprint);

return slprint;

}

void printdata(bst \*root)

{

bst \*run;

run = root;

if(run != NULL)

{

printdata(run->left);

printf("====================================================================\n");

printf("%-7d%-15s %-15s%",run->data.id,run->data.fname,run->data.lname);

if(toupper(run->data.sex) == 'F')

printf("Female");

else

printf("Male ");

printf("%10d",run->data.pay);

if(toupper(run->data.emptype) == 'D')

printf(" Day \n");

else

printf(" Month\n");

printdata(run->right);

}

}

void printfile(bst \*root,FILE \*out)

{

bst \*run;

run = root;

if(run != NULL)

{

printfile(run->left,out);

fprintf(out,"====================================================================\n");

fprintf(out,"%-7d%-15s %-15s%",run->data.id,run->data.fname,run->data.lname);

if(toupper(run->data.sex) == 'F')

fprintf(out,"Female");

else

fprintf(out,"Male ");

fprintf(out,"%10d",run->data.pay);

if(toupper(run->data.emptype) == 'D')

fprintf(out," Day \n");

else

fprintf(out," Month\n");

printfile(run->right,out);

}

}

void printslip(bst \*root)

{

bst \*run;

run = root;

if(run != NULL)

{

printslip(run->left);

printf("====================================================================\n");

printf("%-7d%-15s %-15s%",run->data.id,run->data.fname,run->data.lname);

if(toupper(run->data.sex) == 'F')

printf("Female");

else

printf("Male ");

printf("%10d",run->data.pay);

if(toupper(run->data.emptype) == 'D')

printf(" Day \n");

else

printf(" Month\n");

printf("------------------------ Positive list ------------------------\n");

printf("Day of work(30) = %d, OT(hrs) = %d\n",run->data.dayw,run->data.ot);

printf("Work income = %10.2f, OT income = %10.2f\n",run->data.income,run->data.netot);

if(toupper(run->data.emptype) == 'M')

{

printf("------------------------ Negative list ------------------------\n");

printf("Quota of holidays = 2 day, leave working = %d day\n",run->data.dayh);

printf("Reduction of salary = %.2f\n",run->data.deduction);

}

printf("Net income = %.2f\n",run->data.net);

printslip(run->right);

}

}

//=============================================================================================================================

int start()

{

int sl=-1;

printf("==================================================\n");

printf(" [1] Input employee details\n");

printf(" [2] Input day of work\n");

printf(" [3] Change pay employee and Delete employee\n");

printf(" [4] Calculate Personal Income\n");

printf(" [5] Print salary slip\n");

printf(" [0] Exit Program\n");

printf("==================================================\n");

printf(" Select Menu: ");

scanf("%d",&sl);

return sl;

}

int main()

{

FILE \*out;

out = fopen("printout.txt","a");

bst \*root = NULL;

int cnt=0,sl=-1,slinput=0,slprint=0,sldayw=0,slchange=0,day,i=1;

printf(" Welcome to BU Salary\n");

while(sl != 0)

{

sl = start();

if(sl == 1) //[1] Input employee details

{

slinput = inputmenu();

if(slinput == 1)

{

createbst(&root,&cnt);

sl=-1;

}

else if(slinput == 2)

{

inputkeyboard(&root,&cnt);

sl=-1;

}

else

{

printf("wrong menu please try again !!!\n");

sl=-1;

}

}

else if(sl == 2) //[2] Input day of work

{

sldayw = daywmenu();

if(sldayw == 1)

{

filedayw(root);

printf("\nInput dayw and OT complete !!!\n");

sl=-1;

}

else if(sldayw == 2)

{

inputdayw(root);

sl=-1;

}

else

{

printf("wrong menu please try again !!!\n");

sl=-1;

}

}

else if(sl == 3) //[3] Change pay of employee

{

slchange = changemenu();

if(slchange == 1)

{

root = changeP(root);

sl=-1;

}

else if(slchange == 2)

{

deleteBST(&root,&cnt);

sl=-1;

}

}

else if(sl == 4) //[4] Calculate Persanal Income Tax

{

root = calculatepay(root);

printf("\nCalculate complete!!!!\n");

sl=-1;

}

else if(sl == 5) //[5] Print salary slip

{

slprint = printmenu();

if(slprint == 1)

{

printfile(root,out);

fprintf(out,"Total emp = %d\n",cnt);

printf("\nPrint to file complete !!!\n");

sl=-1;

}

else if(slprint == 2)

{

printf("------------------------- Employee Details -------------------------\n");

printdata(root);

printf("--------------------------------------------------------------------\n");

printf("Total emp = %d\n",cnt);

sl=-1;

}

else if(slprint == 3)

{

printslip(root);

printf("Total emp = %d\n",cnt);

sl=-1;

}

else

{

printf("wrong menu please try again !!!\n");

sl=-1;

}

}

}

printf("See you agin. Good bye!!!\n");

fclose(out);

return 0;

}