if (likePizza == 1) {

printf("I like pizza\n");

}

else {

printf("I hate pizza\n");

printf("I don't want pizza\n");

}

if (age > 12 && age < 16)

printf("Student Fare - no id required\n");

else if (age > 15 && age < 20)

printf("Student Fare - id is required\n");

else if (age < 13)

printf("Child ride for free!\n");

else if (age >= 65)

printf("Senior Fare - id is required\n");

else

printf("Adult Fare\n");

slices = 4;

while (slices > 0) {

slices--;

printf("Gulp! Slices left %d\n", slices); }

slices = 4;

do {

slices--;

printf("Gulp! Slices left %d\n", slices);

} while (slices > 0);

slices = 12;

do {

slices--;

printf("Gulp! Slices left %d\n", slices);

} while (slices < 5);

for (slices = 4; slices > 0; --slices)

printf("Gulp! Slices left %d\n", slices - 1);

switch (choice) {

case 'a':

cost = 1.50; break;

case 'b':

cost = 1.10; break;

default:

choice = '?';

cost = 0.0; }

int main(void) {

printf("minutes left ? ");

scanf("%d", &minutes);

s = minutes > 1 ? 's' : ' ';

printf("%d min%c left\n", minutes, s); return 0; }

[ 5 minutes left

1 minute left ]

int done = 0; // flag

for (i = 0; i < 10 && done == 0; i++) {

printf ("0 to stop");

scanf ("%d", &value);

if (value == 0)

done = 1;

return 0; }

void countDown(int n) {

while (n > 0) {

printf("%d ", n);

n--; }

return; }

void alphabet(void) {

char letter = 'A';

do {

printf("%d ", letter);

letter++;

} while (letter != 'Z')

return; }

void day\_to\_dm (int day, int \*d, int \*m) {

if (day < 32) {

\*m = 1;

\*d = day; }

else if (day < 60)

\*m = 2;

\*d = day - 31; } }

int main(void) {

int day, d, m;

printf("Day of Year : ");

scanf("%d", &day);

day\_to\_dm(day, &d, &m);

printf("Day/Month is %d/%d\n", d, m);

return 0; }

#define SIZE 4

// Declare Struct Employee

struct emp {

int id;

int age;

double salary; };

int main(void) {

struct emp company[SIZE] = { { 0 } };

do {

switch (option) {

case 0: // Quit program

break;

case 1: // Display Employee Data

printf("EMP ID EMP AGE EMP SALARY\n");

for (m = 0; m < num; m++){

printf("%6d%9d%11.2lf\n", company[m].id, company[m].age, company[m].salary);}

break;

case 2:// Adding Employee

if (num < SIZE) {

printf("Enter ID: ");

scanf("%d", &company[num].id);

printf("Enter Age: ");

scanf("%d", &company[num].age);

printf("Enter e Salary: ");

scanf("%lf", &company[num].salary);

num++; }

// If limits are met, display error message

else if (num >= SIZE) {

printf("ERROR!!! Max Reached\n");}

break;

case 3: //Update employee Salary

printf("Update Infon\n");

printf("Enter ID: ");

scanf("%d", &empId);

for (l = 0; l < SIZE; l++) {

if (copany[l].id == empId) {

printf("Enter Employee New Salary: ");

scanf("%lf", &company[l].salary);

printf("\n"); }}

break;

case 4://Remove employee

printf("Enter ID: ");

scanf("%d", &empId);

int index;

for (l = 0; l < num; l++) {

if (copany[l].id == empId) {

index = l;

num--; } }

for (j = index; j < num; j++) {

company[j] = company[j + 1]; }

break;

default:

printf("ERROR: Incorrect Option: Try Again\n"); }

} while (option != 0);

return 0; }

void internal\_swap (int a, int b) {

int c;

printf("a is %d, b is %d\n", a, b);

c = a;

a = b;

b = c; }

internal\_swap(a, b);

OR

int main(void) {

int a, b;

printf("a is ");

scanf("%d", &a);

printf("b is ");

scanf("%d", &b);

printf("a is %d at %x, b is %d at %x\n", a, &a, b, &b);

int c = 0;

c = b; b = a; a = c;

printf("After swap: \n");

printf("a is %d at %x, b is %d at %x\n", a, &a, b, &b);

return 0; }

void swap(int \*p, int \*q) {

int c;

c = \*p;

\*p = \*q;

\*q = c; }

int main(void) {

int a, b;

printf("a is ");

scanf("%d", &a);

printf("b is ");

scanf("%d", &b);

swap(&a, &b);

printf("After swap:\na is %d\n" "b is %d\n", a, b);

return 0; }

#include "student.h"

//Separate header file

struct student {

char first;

int id;

double grade; };

#include <stdio.h>

int main(void) {

struct student Nikki [ ] = {

{'H', 101, 78.5 },{ 'J', 280, 7.89 },{ 'N', 975, 5.89 } };

const int n = 3;

int i;

for (i = 0; i < n; i++)

printf("First initial: %c, ID: %d, mark: %.1lf\n", Nikki[i].first, Nikki[i].id, Nikki[i].grade); }

return 0; }

int power(int base, int exponent) {

int i, result;

result = 1;

for (i = 0; i < exponent; i++)

result = result \* base;

return result; }

int main(void) {

int base, exp, answer;

printf("base? : ");

scanf("%d", &base);

printf("exponent?: ");

scanf("%d", &exp);

answer = power(base, exp);

printf("%d^%d = %d\n", base, exp, answer); }

int main(void) {

int x;

printf("Enter x : ");

scanf("%d", &x);

printf("Value stored in x :%d\n", x);

printf("Address of x :%x\n", &x);

return 0; }

int main(void) {

int i;

int sku[] = { 2156, 4633, 3122, 5611 };

double price[] = { 2.34, 7.89, 6.56, 9.32 };

const int n = 4;

printf(" SKU Price\n");

for (i = 0; i < n; i++)

printf("%5d $%.2lf\n", sku[i], price[i]);

return 0; }

const int NCHAR = 16;

char name[NCHAR + 1] = "My Name is Arnold"

printf("%s", name);

int main(void) {

//declare variables, arrays and initialize them

int numDay = 0;

int avgDay = 0;

int i, j;

int low[10] = { 0 };

int high[10] = { 0 };

int dayHigh = 0, dayLow = 0;

//Prompt greeting message and scan in input from user

printf("---=== IPC Temperature Calculator V2.0 ===---\n");

printf("Please enter the number of days, between 3 and 10, inclusive: ");

scanf("%d", &numDay);

printf("\n");

//Check if conditions are met

while (numDay < 3 || numDay >10) {

printf("Invalid entry, please enter a number between 3 and 10, inclusive: ");

scanf("%d", &numDay);

printf("\n"); }

//Using for loop to scan in high and low temperature values for each day

for (i = 0; i < numDay; i++) {

printf("Day %d - High: ", i + 1);

scanf("%d", &high[i]);

printf("Day %d - Low: ", i + 1);

scanf("%d", &low[i]); }

//Print output using another for loop

printf("\n");

printf("Day Hi Low\n");

for (j = 0; j < numDay; j++) {

printf("%d%d %d\n", j + 1, high[j], low[j]); }

//Initialize the highest and lowest values

int highest = high[0];

int lowest = low[0];

//This loop using if statement will compare and assign the highest values of all temperature inputs plus records the day it falls into

int l = 0;

for (l = 0; l < numDay; l++) {

if (high[l] >= highest) {

highest = high[l];

dayHigh = l + 1; }

if (low[l] <= lowest) {

lowest = low[l];

dayLow = l + 1; }}

printf("The highest temperature was %d, on day %d\n", highest, dayHigh);

printf("The lowest temperature was %d, on day %d\n", lowest, dayLow);

//The check variable done will set a loop until the user wants to stop

int done = 1;

while (done == 1) {

//prompt input number of days from user to calculate average temperature

printf("Enter a number between 1 and 4 to see the average temperature for the entered number of days, enter a negative number to exit: ");

scanf("%d", &avgDay);

printf("\n");

//conditions check

//If conditions are failed, prompt for re-input days

if (avgDay > 4) {

printf("Invalid entry, please enter a number between 1 and 4, inclusive: ");

scanf("%d", &avgDay);

printf("\n"); }

//Check conditions

if (avgDay >= 1 && avgDay <= 4) {

int k;

double average = 0;

int totalHigh = 0, totalLow = 0;

for (k = 0; k < avgDay; k++) {

totalHigh += high[k];

totalLow += low[k]; }

//calculate average

average = ((double)totalHigh + (double)totalLow) / (avgDay \* 2);

printf("The average temperature up to day %d is: %.2lf\n", avgDay, average); }

if (avgDay < 1) {

done = 0; } }

return 0; }

int main(void) {

int x;

int \*apple = &x; // store address of x in p

//Here we let the variable name apple store a value of x

//Since int \*apple is storing the address of variable x by definition (int \*apple = &x)

//When use apple in printf as apple without \*, it show the address

//When use \*apple in printf, it show the value in which the address is holding, this is equivalent to show apple = x;

//Another way to display address of variable x is the following line

//printf("Address of x is: %x\n", &x);

//Use the following to display address of pointer

//printf("Address of pointer stored in pointer: %p\n", apple);

printf("Enter x : ");

scanf("%d", &x);

printf("Value stored in x : %d\n", \*apple); //Show values of x

printf("Address of x : %x\n", apple);

printf("Enter x : ");

scanf("%d", &x);

printf("Value stored in x : %x\n", \*apple); //Also show values of x although %x != %d ??

printf("Address of x : %x\n", apple);

printf("Enter x : ");

scanf("%d", &x);

printf("Value stored in x : %d\n", apple); //Show address of variable x in decimal

printf("Address of x : %x\n", apple);

printf("Enter x : ");

scanf("%d", &x);

printf("Value stored in x : %x\n", apple); //Show address of variable x in hexadecimal

printf("Address of x : %x\n", apple);

return 0; }