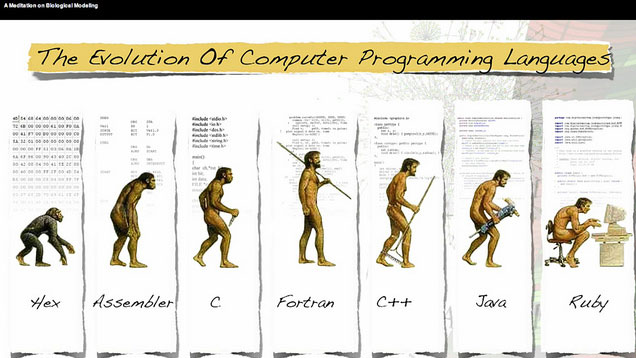
***COP2270***

***M/W***

***Spring 2017-2018***



***Professor: Yassin Raef***

***Anaisy Garcia***

***Chapter 4 Homework/ Exercises***

***Duplicate Figures 4.1, 4.2, 4.5, 4.6, 4.7, 4.9, 4.11, 4.12***

***Problem 4.23 ,4.26 ,4.28 ,4.31***

***One problem per page Please***

***Figure 4-1***

|  |
| --- |
| ***Commands*** |
| ***// Fig. 4.1: fig04\_01.c***  ***// Counter-controlled iteration.***  ***#include <stdio.h>***  ***int main(void)***  ***{***  ***unsigned int counter = 1; // initialization***    ***while(counter <= 10){ // iteration condition***  ***printf("%u\n", counter);***  ***++counter; //increment***  ***}***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Figure 4-2***

|  |
| --- |
| ***Commands*** |
| ***// Fig. 4.2; fig04\_02.c***  ***// Counter-controlled iteration with the for statement.***  ***#include <stdio.h>***  ***int main(void)***  ***{***  ***// initialization, iteration condition, and increment***  ***// are all included in the statement header.***  ***for (unsigned int counter = 1; counter <= 10; ++counter) {***  ***printf("%u\n", counter);***  ***}***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Figure 4-5***

|  |
| --- |
| ***Commands*** |
| ***// Fig. 4.5: fig04\_05.c***  ***// Summation with for.***  ***#include <stdio.h>***  ***int main(void)***  ***{***  ***unsigned int sum = 0; // initialize sum***  ***for (unsigned int number = 2; number <= 100; number += 2){***  ***sum += number; // add number to sum***  ***}***  ***printf("Sum is %u\n", sum);***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Figure 4-6***

|  |
| --- |
| ***Commands*** |
| ***// Fig. 4.6: fig04\_06.c***  ***// Calculating compound interest.***  ***#include <stdio.h>***  ***#include <math.h>***  ***int main(void)***  ***{***  ***double principal = 1000.0; // starting principal***  ***double rate = .05; // annual interest rate***  ***//output table column heads***  ***printf("%4s%21s\n", "Year", "Amount on deposit");***  ***// calculate amount on deposit for each of ten years***  ***for (unsigned int year = 1; year <= 10; ++year){***  ***// calculate new amount for specified year***  ***double amount = principal \* pow(1.0 + rate, year);***  ***// output one table row***  ***printf("%4u%21.2f\n", year, amount);***  ***}***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Figure 4-7***

|  |
| --- |
| ***Commands*** |
| ***//Fig. 4.7: fig04\_07.c***  ***//Counting letter grades with switch.***  ***#include <stdio.h>***  ***int main(void)***  ***{***  ***unsigned int aCount = 0;***  ***unsigned int bCount = 0;***  ***unsigned int cCount = 0;***  ***unsigned int dCount = 0;***  ***unsigned int fCount = 0;***  ***puts("Enter the letter grades.");***  ***puts("Enter the EOF character to end input.");***  ***int grade; // one grade***  ***// loop until user types end-of-file key sequence***  ***while ((grade = getchar()) != EOF) {***  ***//determine which grade was input***  ***switch (grade) { // switch nested in while***  ***case 'A': // grade was uppercase A***  ***case 'a': // or lowercase a***  ***++aCount;***  ***break; //necessary to exit switch***  ***case 'B': //grade was uppercase B***  ***case 'b': // or lowercase b***  ***++bCount;***  ***break;***  ***case 'C': // grade was uppercase C***  ***case 'c': // or lowercase c***  ***++cCount;***  ***break;***  ***case 'D': // grade was uppercase C***  ***case 'd': // or lowercase c***  ***++dCount;***  ***break;***  ***case 'F': // grade was uppercase C***  ***case 'f': // or lowercase c***  ***++fCount;***  ***break;***  ***case '\n': // ignore newlines,***  ***case '\t': // tabs,***  ***case ' ': // and spaces in input***  ***break;***  ***default: // catch all other characters***  ***printf("%s", "Incorrect letter grade entered.");***  ***puts("Enter a new grade.");***  ***break; // optional; will exit switch anyway***  ***}***  ***} // end while***  ***// output summary of results***  ***puts("\nTotals for each letter grade are:");***  ***printf("A: %u\n", aCount);***  ***printf("B: %u\n", bCount);***  ***printf("C: %u\n", cCount);***  ***printf("D: %u\n", dCount);***  ***printf("F: %u\n", fCount);***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Figure 4-9***

|  |
| --- |
| ***Commands*** |
| ***// Fig. 4.9: fig04\_09.c***  ***// Using the do...while iteration statement.***  ***#include <stdio.h>***  ***int main(void)***  ***{***  ***unsigned int counter = 1; // initialize counter***  ***do{***  ***printf("%u ", counter);***  ***} while (++counter <= 10);***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Figure 4-11***

|  |
| --- |
| ***Commands*** |
| ***// Fig. 4.11: fig04\_11.c***  ***// Using the break statement in a for statement.***  ***#include <stdio.h>***  ***int main(void)***  ***{***  ***unsigned int x; // declared here so it can be used after loop***  ***// loop 10 times***  ***for(x = 1; x <= 10; ++x){***  ***// if x is 5, terminate loop***  ***if (x == 5) {***  ***break; // break loop only if x is 5***  ***}***  ***printf("%u ", x);***  ***}***  ***printf("\nBroke out of loop at x == %u\n", x);***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Figure 4-12***

|  |
| --- |
| ***Commands*** |
| ***// Fig. 4.12: fig04\_12.c***  ***// Using the continue statement in a for statement***  ***#include <stdio.h>***  ***int main(void)***  ***{***  ***// loop 10 times***  ***for (unsigned int x = 1; x <= 10; ++x) {***  ***// if x is 5, continue with next iteration of loop***  ***if (x == 5) {***  ***continue; // skip remaining code in loop body***  ***}***  ***printf("%u ", x);***  ***}***  ***puts("\nUsed continue to skip printing the value 5");***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Problem 4-23***

|  |
| --- |
| ***Commands*** |
| ***// Fig 4.23: fig04\_23.c***  ***// Calculating the compound interest with integers***  ***#include <stdio.h>***  ***#include <math.h>***  ***int main(void)***  ***{***  ***unsigned int year;***  ***unsigned int amount;***  ***unsigned int dollars;***  ***unsigned int cents;***  ***unsigned int principal = 100000;***  ***double rate = .05;***  ***printf("%s%21s\n", "year", "Amount on deposit");***  ***for(year = 1; year <= 10; ++year) {***  ***amount = principal \* pow(1.0 + rate, year);***  ***cents = amount % 100; // takes cents the last two digits***  ***dollars = amount / 100; // emilinates decimals***  ***printf("%4u%18u", year, dollars); // display year followed by a period***  ***if (cents < 10) {***  ***printf("0%u\n", cents); // puts a zero and a cent***  ***} // end if***  ***else {***  ***printf("%u\n", cents); // if not put cents***  ***} // end else***  ***} // end for***  ***} // end main*** |

|  |
| --- |
| ***Output*** |
|  |

***Problem 4-26***

|  |
| --- |
| ***Commands*** |
| ***// Fig 4.26: fig04\_26.c***  ***// Calculating the value of pi***  ***# include <stdio.h>***  ***int main(void)***  ***{***  ***double pi = 0.0; // approximated pi***  ***double numb = 4.0; // numerator***  ***double denumb = 1.0; // denominator***  ***unsigned int loop; // looper***  ***unsigned int accuracy; // number of terms***  ***accuracy = 100000; // number of time loop terms occur***  ***// display headers***  ***printf("Accuracy set at: %u\n", accuracy);***  ***puts ("Term\t\t pi");***  ***// create a looper***  ***for(loop = 1; loop <= accuracy; ++loop) {***  ***//TEST IF ODD OR EVEN***  ***// test for odd value***  ***if(loop % 2 != 0){***  ***pi += numb / denumb; // if odd, add to the current pi value the numerator over the denominator***  ***} // end if***  ***// if not odd must be even***  ***else{***  ***pi -= numb / denumb; // if even, subtract to the current pi value the numerator over the denominator***  ***} // end else***  ***// display number of terms and approximated vlue for pi***  ***printf("%u\t\t %f\n", loop, pi);***  ***denumb += 2.0; // update denominator by adding two***  ***} // end for***  ***} // end main*** |

|  |
| --- |
| ***Output*** |
|  |

***Problem 4-28***

|  |
| --- |
| ***Commands*** |
| ***// Fig 4.28: fig:04\_28.c***  ***// Calculating Weekly Pay***  ***#include <stdio.h>***  ***int main (void)***  ***{***  ***int pieces;***  ***int code;***  ***float pay;***  ***float sales;***  ***float hours;***  ***float total = 0;***  ***printf("\nEnter employee's paycode (-1 to end): ");***  ***scanf("%d", &code);***  ***while (code != -1) {***  ***switch (code) {***  ***case 1: // manager***  ***printf("\nEnter manager's pay rate: ");***  ***scanf("%f", &pay);***  ***printf("Weekly pay is: %.2f\n\n", pay);***  ***total += pay;***  ***break;***  ***case 2: // hourly worker***  ***printf("\nEnterly hourly worker's pay rate: ");***  ***scanf("%f", &pay);***  ***printf("Enter the number of hours worked: ");***  ***scanf("%f", &hours);***  ***if (hours > 40)***  ***pay = (pay \* 40) + ((hours - 40) \* (pay \* 1.5));***  ***else***  ***pay \*= hours;***  ***printf("Weekly pay is: %.2f\n\n", pay);***  ***total += pay;***  ***break;***  ***case 3: // commision workers***  ***printf("\nEnter commission employee's weekly sales: ");***  ***scanf("%f", &sales);***  ***pay = 250 + (.057\* sales);***  ***printf("Weekly pay is: %.2f\n\n", pay);***  ***total += pay;***  ***break;***  ***case 4: // pieceworker***  ***printf("\nEnter number of items produced: ");***  ***scanf("%d", &pieces);***  ***printf("Enter number per item pay rate: ");***  ***scanf("%f", &pay);***  ***pay = pieces \* pay;***  ***printf("Weekly pay is: %.2f\n\n", pay);***  ***total += pay;***  ***break;***  ***default: // for invalid paycode***  ***printf("Invalid Code Entered.\n");***  ***}***  ***printf("\nEnter employee's paycode (-1 to end): ");***  ***scanf("%d", &code);***  ***}***  ***printf("\nThe total payroll for the week is: %.2f\n", total);***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

***Problem 4-31***

|  |
| --- |
| ***Commands*** |
| ***// Fig.04.31: fig04\_21.c***  ***// Diamond-Printing Program***  ***#include <stdio.h>***  ***int main(void)***  ***{***  ***int line;***  ***int space; // " "***  ***int asterisk; // \****  ***// top triangle***  ***for (line = 1; line <= 9; line += 2) {***  ***for (space = (9 - line) / 2; space > 0; space --) {***  ***printf(" ");***  ***}***  ***for (asterisk = 1; asterisk <= line; asterisk++) {***  ***printf("\*");***  ***}***  ***printf("\n");***  ***}***  ***// bottom flipped triangle***  ***for (line = 7; line >= 0; line -= 2) {***  ***for (space = (9 - line) / 2; space > 0; space--) {***  ***printf(" ");***  ***}***  ***for (asterisk = 1; asterisk <= line; asterisk++) {***  ***printf("\*");***  ***}***  ***printf("\n");***  ***}***  ***}*** |

|  |
| --- |
| ***Output*** |
|  |

|  |  |
| --- | --- |
| ***Command(s) learned*** | |
| ***command*** | ***comment*** |
| ***++a*** | ***Increment by 1 then use new value*** |
| ***--a*** | ***Use current value and increment by 1*** |
| ***a++*** | ***Decrement by 1 then use new value*** |
| ***a--*** | ***Use current value and decrement by 1*** |
| ***for*** | ***A type of loop with repetition control which will loop a specific amount of times*** |
| ***+=*** | ***A += B; A = A + B*** |
| ***-=*** | ***A -= B; A = A - B*** |
| ***!=*** | ***Not equal to*** |
| ***Unsigned int*** | ***Integers that can only be positive*** |
| ***#include <math.h>*** | ***Find the math library allows for more complicated mathematics*** |
| ***double*** | ***Double is a bigger version of float meaning it can hold more floating-point numbers*** |
| ***pow*** | ***Written as( double pow(double x, double y); X being a base and Y being a power; raises x to the power of y*** |
| ***%s*** | ***Take the argument and print it as a string*** |
| ***%u*** | ***Take the argument and print it as a unsigned decimal integer*** |
| ***getchar*** | ***Gets a character*** |
| ***EOF*** | ***End of file; has value -1; no more data; can press control Z or D to cause an end of file*** |
| ***<=*** | ***Less than or equal to*** |
| ***>=*** | ***Greater than or equal to*** |
| ***break*** | ***Can terminate a loop and continue to the next statement; can terminate a case in switch*** |
| ***case*** | ***Goes within switch; used to present different statements*** |
| ***do*** | ***Like while loop but guaranteed to execute; checks condition at bottom of loop*** |
| ***default*** | ***Can be used if none of the cases within switch are true*** |
| ***switch*** | ***Faster than if and else statements; allows a variable to be tested against a list of values*** |