**PART 1:**

#include <stdio.h>

#include <ctype.h>

int main (){

char c;

printf("enter a number or letter.\n\n : ");

scanf("%c", &c);

printf("\n\n");

while (c!='z' && c!='Z'){

if (isalpha(c))

printf("DING\n\n");

if (isdigit(c))

printf("DONG\n\n");

printf("enter a number or letter.\n\n : ");

scanf("\n%c", &c);

printf("\n\n");

}

printf("The zed word!\n\n");

return 0;

}

**Part 2:**

#include <stdio.h>

#include <ctype.h>

typedef struct animal {

int age;

int weight;

int units;

char name[100];

} Animal;

//------------------------------------------------------------------------------------------------Fill Form Function

void fill\_form(int \*agePtr, int \*weightPtr, int \*unitsPtr, char \*namePtr){

char trash[5];

int digiCheck;

printf("%14s: ","age");

scanf("%d", agePtr);

fgets(trash, 4, stdin);

digiCheck= \*agePtr;

while (\*agePtr>=100 || \*agePtr<0){

printf("Invalid value, please input a number between 0 and 100\n");

printf("%14s: ","age");

scanf("%d", agePtr);

fgets(trash, 4, stdin);

digiCheck= \*agePtr;

}

printf("%14s: ","weight (lbs)");

scanf("%d", weightPtr);

fgets(trash, 4, stdin);

digiCheck= \*weightPtr;

while (\*weightPtr>999999 || \*weightPtr<0){

printf("Invalid value, please input a number between 0 and 999,999\n");

printf("%14s: ","weight (lbs)");

scanf("%d", weightPtr);

fgets(trash, 4, stdin);

digiCheck= \*weightPtr;

}

printf("%14s: ","units");

scanf("%d", unitsPtr);

fgets(trash, 4, stdin);

digiCheck= \*unitsPtr;

printf("%14s: ","name");

fgets(namePtr, 99, stdin);

}

//------------------------------------------------------------------------------------------------Read Form Function

void read\_form(int \*agePtr, int \*weightPtr, int \*unitsPtr, char \*namePtr){

int voidCounter=0;

printf("%14s: %d\n","age", \*agePtr);

printf("%14s: %d\n","weight (lbs)", \*weightPtr);

printf("%14s: %d\n","units", \*unitsPtr);

printf("%14s: ", "name");

while (namePtr[voidCounter]!='\0' ){

printf("%c", namePtr[voidCounter]);

voidCounter++;

}

printf("\n");

}

//------------------------------------------------------------------------------------------------Read Switch Operator

int switchOperator( int total){

int voidCounter;

int returnVal;

char trash[5];

printf("which do you want?\n");

for (voidCounter=1; voidCounter<=total; voidCounter++){

if (voidCounter%2==0)

printf("%14s (%d)\n", "animal", voidCounter);

else

printf("%14s (%d)", "animal", voidCounter);

}

printf("%14s (0)\n\n : ", "All");

scanf("%d", &returnVal);

fgets(trash, 4, stdin);

return returnVal;

}

//------------------------------------------------------------------------------------------------Main Function

int main (){

int counter;

char readChar;

char writeChar;

int readSwitchLever;

int writeSwitchLever;

int animalCount;

printf("how many animals would you like to document?\n\n : ");

scanf("%d", &animalCount);

Animal array[animalCount];

char trash[5];

fgets(trash, 4, stdin);

for (counter=0; counter<animalCount; counter++){

printf("\n (animal number: %d)\n\n", (counter+1));

fill\_form(&array[counter].age, &array[counter].weight, &array[counter].units, array[counter].name);

}

printf("Press \"y\" if you would like to review the forms. \n : ");

scanf("%1c", &readChar);

fgets(trash, 4, stdin);

readChar=toupper(readChar);

while (readChar=='Y'){

readSwitchLever= switchOperator(animalCount);

switch (readSwitchLever){

case 0: printf("\n\n");

counter=0;

while (counter<animalCount){

printf("\n (animal number: %d)\n\n", (counter+1));

read\_form(&array[counter].age, &array[counter].weight, &array[counter].units, array[counter].name);

counter++;

}

break;

default:

printf("\n\n");

read\_form(&array[readSwitchLever-1].age, &array[readSwitchLever-1].weight, &array[readSwitchLever-1].units, array[readSwitchLever-1].name);

break;

}

printf("Press \"y\" if you would like to edit the forms \n : ");

scanf("%1c",&writeChar);

fgets(trash, 4, stdin);

writeChar= toupper(writeChar);

if (writeChar=='Y'){

writeSwitchLever= switchOperator(animalCount);

switch (writeSwitchLever){

case 0: printf("\n\n");

counter=0;

while (counter<animalCount){

printf("\n (animal number: %d)\n\n", (counter+1));

fill\_form(&array[counter].age, &array[counter].weight, &array[counter].units, array[counter].name);

counter++;

}

break;

default:

printf("\n\n");

fill\_form(&array[writeSwitchLever-1].age, &array[writeSwitchLever-1].weight, &array[writeSwitchLever-1].units, array[writeSwitchLever-1].name);

break;

}

}

printf("Press \"y\" if you would like to review the forms, again. \n : ");

scanf("%c", &readChar);

fgets(trash, 4, stdin);

readChar= toupper(readChar);

}

return 0;

}

**Part 3:**

Aside from knowing the objects being used and referenced in the code, comment lines and spacing are probably the most decisive factors in readability. It is convenient when the variable declarations are all at the top of a function rather than scattered throughout a function. I like to see comment lines disambiguate obscure variable names and explain, or at least coherently label, major objects like function declarations/ definitions, structure definitions, etc.

While spacing preference is personal, I find it easiest to read when the code block within each control flow structure is indented 1 tab’s worth farther than the structure itself. Also, lines that perform similar or closely related tasks be single spaced, and each of such related blocks separated by double, or even triple spacing.