My Answers

Yeah, this ain’t a “How to Pass”, exactly. But it is the simplest (not the best or most efficient, *Matt*) answers to a bunch of exam questions, 3 of which repeated (C, E and F are the same in both years, more or less).

Anyway, if you have any questions, ask.

I’ve also given the questions star ratings with for difficulty, with higher stars being harder, so bear that in mind when you’re deciding what to learn.

Also, B’s seem to be screwed. Summer 2023 is indecipherable for me, and Autumn 2023 is just awkward (I used weird syntax to make it less so, but it’s not worth your learning).

Anyhoodles, here’s a bunch of code:  
  
2023 – Summer

#include <stdlib.h>

#include <stdio.h>

#include <ctype.h>

//This is for the 2022/2023 Summer past paper

//Difficulty is rated out of 5 stars (\*), with 5 stars (\*\*\*\*\*) being the max difficulty

//A. \*

void FibonnaciCalc(int numOfPrints) {

int fibNum = 1;

int prevNum = 0;

for(int i = 0; i< numOfPrints; i++){

printf("%d ", fibNum);

int nextFib = fibNum+prevNum;

prevNum = fibNum;

fibNum = nextFib;

}

}

//B. ?

//This question was kinda just nonsense, you should avoid

//C. \*\*\*

int countEs(char string[100]){

int size = 100;

int numOfE = 0;

for(int i =0; i<size; i++){

if(isalpha(string[i])){

if(string[i] == 'e' || string[i] == 'E'){

numOfE = numOfE + 1;

}

}

else{printf("%c ", string[i]);}

}

return(numOfE);

}

//D. \*\*\*\*

void printDate(int daysSinceStart){

int monthNum = 0;

char \*monthNames[]= {"January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"};

int monthSizes[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

while (daysSinceStart > monthSizes[monthNum]){

daysSinceStart -= monthSizes[monthNum];

monthNum++;

}

printf("%s %d", monthNames[monthNum], daysSinceStart);

}

//E. \*\*

int findEvenThreeNums(){

int sum = 0;

int num = 300;

int maxNum = 6000;

while(num<=6000){

if(num % 2 == 0 && num % 3 == 0){

sum += num;

}

num++;

}

for(num; num<=6000; num++){

if(num % 2 == 0 && num % 3 == 0){

sum += num;

}

}

return(sum);

}

//F. \*

void findSum(int firstInt, int secondInt){

int sum = 0;

for(int i=firstInt+1; i<secondInt; i++){

sum+=i;

}

printf("%d", sum);

}

int main(){

// FibonnaciCalc(20);

// char eString[100] = {"zAu5xg8gn&HM,;M7;V!S8a7EgdeqTLyeeeeNFFt:x7[\*b-tCRZJTi[Qje/6fv!%XyN/XngfcLx4/10V1c}#eGLwbtM&j:j6ZpW1,"};

// printf("\n");

// int numE = countEs(eString);

// printf("\n%d", numE);

//printDate(133);

// int sum = findEvenThreeNums();

// printf("%d", sum);

findSum(3, 7);

}

# 2023 – Autumn

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

//A. \*

float doubleTo25(float num){

int count = 0;

for(count; num<=25; count++){num=num\*2;}

printf("%f ", num);

return(count);

}

//B. \*\*\*\*

void operations(float num1, float num2, float num3){

float biggest, middlest, smallest;

biggest = (num1>num2) ? ((num1>num3) ? num1:num3) :((num2>num3) ? num2:num3);

smallest = (num1<num2) ? ((num1<num3) ? num1:num3) :((num2<num3) ? num2:num3);

middlest = (num1!=biggest && num1!=smallest) ? num1 : ((num2!=biggest && num2!=smallest) ? num2 : num3);

float product = middlest\*smallest;

float sum = middlest+biggest;

printf("Product of 2 smallest: %f", product);

printf("Sum of 2 largest: %f", sum);

}

//C. \*\*\* - Same as sumemer C

int countX(char string[1000]){

int size = 1000;

int numOfX = 0;

for(int i =0; i<size; i++){

if(isalpha(string[i])){

if(string[i] == 'x' || string[i] == 'X'){

numOfX += 1;

}

}

else if (isnumber(string[i])){

printf("%c ", string[i]);

}

}

return(numOfX);

}

//D. \*\*

void timeCheck(int numSecs){

int hour = 0;

int minute = 0;

while(numSecs>59){

numSecs-=60;

minute++;

}

while(minute>59){

minute-=60;

hour++;

}

printf("%d:%d:%d", hour, minute, numSecs);

}

//E. \* - Same as Summer E

int oddDivis5Sum(){

int sum = 0;

int num = 500;

int maxNum = 5000;

while(num<=5000){

if(num % 2 != 0 && num % 5 == 0){

sum += num;

}

num++;

}

for(num; num<=5000; num++){

if(num % 2 != 0 && num % 5 == 0){

sum += num;

}

}

return(sum);

}

//F. \* - Same as summer F

void findSum(int firstInt, int secondInt){

int sum = 0;

for(int i=firstInt+1; i<secondInt; i++){

sum+=i;

}

printf("%d", sum);

}

int main(){

// int value = doubleTo25(0.000000007);

// printf("%d ", value);

// operations(23.42153, 0.952456, -527524.34);

// int numX = countX("3KvDL6B;BY}t+DxU-5$.:gB8H0wWyW!3h==\_K)-cvQx.u9:LPDcKfR&f3UEV0Z7T+--kyrp.D98[U]$)bL!u7HAFjrztpzb@Vjq8-5GE$dM;dL\*w+Hr}(9JRjG=Q,yMqwe;#&1NK@V0($L[=V#8nyY9.rHu6Qf/+6tq}7t&fuBTieA.e,;UNHJ)UCYQ.g5L!+bc.d\*3-#jj}$r=d.;tHJ,y!yHjWa2v3?G?8.uUu:P{!dmm[a.3;F$[bjm$p?B=CTv5}gQ?!amy-nPC%uLSxG+GWJm1KkvtVFnZ0\*w}hKh9mQkgZrgn0\*V&V{Kx/Pu?;U5zZ[b?Xq[xn=#5G6}RVv+-why?GnuC+ZS]${Rbg&ZK{#-ESAdY-j;@:H[N$#+\_b4F6D:NTP8x#iL&}Z6xQ1c:BTBBj&{];P8vjT@Wgwq%E\*FkX(;0V+1Z\*(k0pnfyZaw,29kZ8/H3twu{eFfWw=])\_uiK.fai}eAv4d;]?%0UnA/avg\_916P:p@aiQq\*ai$v}+J}@km#cwGvR?\*GHxpdmuDA(JL0[Thc()dRFL8AR+ex{P=(XTyw\_bFnkr6+%TX}K3wp7.=7+({pWm$LX56UVbMh!Z7AKFnECbP&9Uc\*g3Qt9@LhP?L{=U\*J42[kK2r2,ppF?CZ?;9XmB:]D:Z]Lu:\*t$aBk{8c5A)$#6d(]7:XaGD@,\_$eHWd8Mm#SYTkUy+cm4,+H]a(C1u90:=U5[?C+Bx]=PdXW&4G5623jq#DJ-t!\*Kpya&k1rE3Ad,.T#9Q1kU&E/?767+7)xKD0NRJLMpek@)\_tM]!+15z\*QLfrSdXLx+[6W::VE=1H7Ugzvy#BS6VtnJtn@u#]6}{gpc7vZq\*4=a0}ptPCq{0V87+Xdy4p3XA3]!wZ,}hffkGP}h:FdeAx!EP#j9!$e,8R@nN!!,ckRB+P?C;Ect[yanUC\_b7fd&\*\*v;zg.iiR;tqN!1wq+r8jz2J}-ZA@(R16gwh98?.q9&F+77S");

// printf("Num of X's: %d", numX);

// timeCheck(42933);

int result = oddDivis5Sum();

printf("%d", result);

}

# 2022 – Summer

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#include <math.h>

//A. \*\*\* - for awkwardness more than difficulty(which would be a \*\*)

int calcVisitors(int visitors1[7], int visitors2[7], int visitors3[7]){

int moreThan1000\_1 = 0;

int moreThan1000\_2 = 0;

int moreThan1000\_3 = 0;

int sum = 0;

for(int i =0; i<7; i++){

if (visitors1[i]>1000){moreThan1000\_1++;}

}

for(int i=0; i<7; i++){

if (visitors2[i]>1000){moreThan1000\_2++;}

}

for(int i=0; i<7; i++){

if (visitors3[i]>1000){moreThan1000\_3++;}

}

for(int i=0; i<7; i++){

sum += visitors1[i]+ visitors2[i] + visitors3[i];

}

printf("No. Times Site 1 Exceeded 1000 Visitors: %d", moreThan1000\_1);

printf("No. Times Site 2 Exceeded 1000 Visitors: %d", moreThan1000\_2);

printf("No. Times Site 3 Exceeded 1000 Visitors: %d", moreThan1000\_3);

return(sum);

}

//B. \*\*\*

int sumEvenDigits(int number){

int sum=0;

for(int i=0; i<20; i++){

if (number!=0){

int digit = number % 10;

if(digit % 2 == 0){

sum+=digit;

}

number = number/10;

}

else{break;}

}

return(sum);

}

//C. \*\*

int checkIntConditions(int integerArray[100]){

int numThatFit = 0;

for(int i=0; i<100; i++){

int number = integerArray[i];

if(number<300){

if(number>75){

if(number % 9 == 0){

if(number % 2 ==0){

numThatFit++;

}

}

}

}

}

return(numThatFit);

}

//D. \*

float volumeCalc(float radius){

float volume = (4/3)\*3.14\*radius\*radius\*radius;

}

// int main(){

// float volume = volumeCalc(57.6)-volumeCalc(33.3);

// printf("Volume: %f", volume);

// }

//E.

float holeVolumeCalc(float length, float height, float depth){

float gravelVolume = 7.23;

float volume = 0.5\*length\*height\*depth;

float deciNumoOfBags = volume/gravelVolume;

float numOfBags = (float)(ceil(deciNumoOfBags));

return(numOfBags);

}

//F.

void halfOddSum(int num1, int num2){

int big, small;

if(num1>num2){big=num1; small=num2;}

else{big=num2; small=num1;}

int sum = 0;

for(int i = small+1; i<big; i++){

if(i%2!=0){

sum+=i;

}

}

int halfSum = sum/2;

printf("%d", halfSum);

}

int main(){

int EvenDigitsSum = sumEvenDigits(49670124);

printf("%d", EvenDigitsSum);

return 0;

}

# 2022 – Autumn

#include <stdio.h>

#include <stdlib.h>

//A. \*

int Factorial(int number){

int factorial = 1;

for(int i=1; i<=number; i++){

factorial = factorial\*i;

}

return(factorial);

}

//B. Omitted because the example was dumb and I didn't want to have to deal with it. You heard it here folks, skip part B's

//C. \*\*

int findingThings(int numbers[100]){

int sum3 = 0;

int numGreaterThan25 = 0;

for(int i =0; i<100; i++){

if(numbers[i]%3==0){

sum3+=numbers[i];

}

if(numbers[i]>25){

numGreaterThan25++;

}

}

printf("%d", numGreaterThan25);

return(sum3);

}

//D. - Same as 2023 D

void timeCheck(int numSecs){

int hour = 0;

int minute = 0;

while(numSecs>59){

numSecs-=60;

minute++;

}

while(minute>59){

minute-=60;

hour++;

}

printf("Hours: %d Minutes: %d Seconds: %d", hour, minute, numSecs);

}

//E. - Same as both 2023 E's

int sumOdds(){

int sum = 0;

int num = 1000;

int maxNum = 1500;

while(num<=maxNum){

if(num % 2 != 0){

sum += num;

}

num++;

}

for(num; num<=maxNum; num++){

if(num % 2 != 0){

sum += num;

}

}

return(sum);

}

//F. \*\*\*

void subtractSmaller(int num1, int num2){

int bigger, smaller, bufferNum;

if(num1>num2){bigger=num1; smaller=num2;}

else{{bigger=num2; smaller = num1;}}

printf("%d %d\n", bigger, smaller);

while(bigger>=1 && smaller>=1){

bigger-=smaller;

if(bigger>=1 && smaller>=1){

if(smaller>bigger){bufferNum=bigger; bigger=smaller; smaller=bufferNum;}

printf("%d %d\n", bigger, smaller);

}

}

}

int main(){

subtractSmaller(2342, 8906);

// int fact = Factorial(5);

// printf("%d", fact);

return(0);

}