

## Hw1-1

```
// This program checks whether user's integer input is a prime number.  
// All rights are reserved to Boris Milner & Max Moldavsky.
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int input,valid,counter;
```

```
        printf("\n\nPlease enter an integer : ");
```

```
        valid=scanf("%d", &input);
```

```
        if (valid<1) {printf("Sorry, the input is invalid.\n");return 1;}
```

```
        if (input<=1) {printf("Sorry, we only deal with integers higher than 1.\n");return 1;}
```

```
                for (counter=2; counter<=input/2; counter++) // The search for divisors...
```

```
                { if (input%counter==0) {printf("The integer is not a prime number, %d is
```

```
a divisor.\n\n\n",counter); return 0;}
```

```
                }
```

```
                printf("The input is a prime number.\n\n\n"); // No divisors found? --> Seems like
```

```
we found a prime number !
```

```
        return 0;
```

```
}
```

## HW1-2

//This is the CORRECT version of this program from our home-work.

//The PREVIOUS one was printed from a wrong file...

//Please Accept our appologies...

// 24-11-2005

// This program explains to you what you have already known about your birthday.

// All rights are reserved to Boris Milner & Max Moldavsky.

```
#include <stdio.h>
```

```
int main()
```

```
{
    int day,month,year;
    printf("\n\nGreetings, the date today is 14/11/2005");
    printf("\n\n\nPlease enter your DAY of birth : ");
    scanf("%d",&day);
    if (day>=1) if (day<=30)
        {printf("\nPlease enter your MONTH of birth : ");
          scanf("%d",&month);
            if (month>=1) if (month<=12)
                {printf("\nPlease enter your YEAR of birth : ");
                  scanf("%d",&year);
                }
            }
        }
```

```

    if ((year<2005) || (year==2005&&month<11) ||
        (year==2005&&month==11&&day<=14))
        {printf("\nYour birthday is on %d/%d/%d
",day,month,year);
          if (day>=14&&month>=11) year=(2005-year-1);
          else (year=2005-year);
          printf("Which means you are %d years
old.",year);
          if (day==14&&month==11) {year=year+1;
          printf("\n\nHappy Birthday dude ! - tomorow you'll be %d years old.",year);}
          return 0;
        }
    }
```

```

    printf("\nSorry, at least one of the values entered is invalid.");
    return 1;
}
```

### HW1-3

```
// This program returns the area of user chosen triangle by using Heron's formula.
// All rights are reserved to Boris Milner & Max Moldavsky.

#include <stdio.h>
#include <math.h>

int main()
{
double a,b,c,p,area;

printf("\n\nPlease enter the length of side A of the triangle : ");
scanf("%lf",&a);
printf("Please enter the length of side B of the triangle : ");
scanf("%lf",&b);
printf("Please enter the length of side C of the triangle : ");
scanf("%lf",&c);
if (a<=0||b<=0||c<=0) {printf("\n\nInput error..."); return 1;} // Can't happen in the world we know.
if (a+b<=c||b+c<=a||c+a<=b) {printf("\n\nLogical error..."); return 1;} // It's not a triangle.

p=0.5*(a+b+c);
area=( sqrt(p*(p-a)*(p-b)*(p-c)) ); // (C) Heron
printf("\nThe area of the triangle above is : %.2lf\n",area);
return 0; // If you liked this program buy it.
}
```

## HW2-1

```
// This program converts European grade system to American one.
// All rights are reserved to Boris Milner & Max Moldavsky.

#include <stdio.h>

int grade,valid;

int main()
{
printf("\n\nPlease enter EUROPEAN grade (MUST be between 0 and 100) : ");
valid=scanf("%d",&grade); // Shouldn't return 1 if input is invalid.
if (valid<1) {printf("Invalid input, exiting...\n"); return(1);}

while (grade<0 || grade>100) // Smartguy punching invalid grades.

    { printf("\n\nIllegal grade (MUST be between 0 and 100), Please try again : ");
      valid=scanf("%d",&grade);
      if (valid<1) {printf("Invalid input, exiting...\n"); return(1);}
    }

    if (grade<=60) {printf("\nAmerican grade is : F\n"); return(0);} // Good luck next time.
    else printf("\nAmerican grade is : %c\n",75-(grade+9)/10); // ASCII value of 'A' can be found
with printf("%d",'A');

return 0;
}
```

## HW2-2

```
// This program shows statistics about a variable-length serie entered.
// All rights are reserved to Boris Milner & Max Moldavsky.

#include <stdio.h>

int length=0,temp=1,min,max,counter,valid,total;

int main()
{
printf("\n\nPlease enter a sequence : ");
valid=scanf("%d",&min); max=total=min; // Validation check. - First input is a multi-value.
if (valid<1) {printf("\nError reading input, the program is aborted.\n"); return(1);}
if (min==0) {printf("\nZero-length sequence!\n"); return(0);}
while (counter=1,temp!=0,counter++) // Compiler sees things as they are and ask why.
// We see things as they will be and say why not.
{
valid=scanf("%d",&temp);
if (valid<1) {printf("\nError reading input, the program is aborted."); return(1);}
if (temp>max) max=temp;
if (temp<min&&temp!=0) min=temp;
length++;
total+=temp;
if (temp==0) break;
}
if (min<0) {printf("\nInvalid input (negative number). Exiting...\n"); return(1);}
printf("\n\nLength : %d",length);
printf("\nMaximum : %d",max);
printf("\nMinimum : %d",min);
printf("\nMean : %lf",(double)total/length); // Considers the division may result in a real number.

return(0);
}
```

## HW2-3

```

// This program shows information about a new Technion bus line.
// All rights are reserved to Boris Milner & Max Moldavsky.

#include <stdio.h>

#define WB 6
#define WE 22

// :: WB --> Work Begins :: WE --> Work Ends ::

int buses=0,shour,sminut,fhour,fminut,valid,counter,plus,temp;
int main() // :: s=start :: f=finish :: plus=time intervals ::
{
    printf("\n\nPlease enter start time (6:00 - 22:58) : ");
    valid=scanf("%d:%d",&shour,&sminut);
    if (valid<2) {printf("Invalid Input, exiting...\n");return(1);} // Validation check.
    printf("\n\nPlease enter finish time (6:01 - 22:59) : ");
    valid=scanf("%d:%d",&fhour,&fminut);
    if (valid<2) {printf("Invalid Input, exiting...\n"); return(1);} // Validation check.
    if (shour<WB || shour>WE || sminut<0 || sminut>59 ||
        fhour<WB || fhour>WE || fminut<0 || fminut>59) {printf("\nIllegal time, exiting...\n"); return(1);}
    if (shour>fhour || (shour==fhour&&fminut<=sminut)) {printf("\nEnd time must be larger than start
time!\n"); return(1);}

    printf("\n\nTotal time : %d:",fhour-shour);
    if ( (fminut-sminut)<10 ) printf("0"); printf("%d\n\n",fminut-sminut);

    for (counter=shour;counter<=fhour;counter++) // Chart creating loop (R)
        {printf("%d",counter); if (counter<10) printf(" "); // Will improve this algorithm in V1.5 ...
            printf(" %d:00",counter);buses++; // Here goes the first bus.
            for (temp=0;temp<60;temp+=plus){
                if (counter<=9) plus=10;
                if ( (counter>=10)&&(counter<=18) )
                    plus=15;
                if ( (counter>=19)&&(counter<=22) )
                    plus=20;
                if (temp>0)
                    {printf("%8d:%d",counter,temp);buses++;} // Now all the other buses.
            }
            printf("\n");
        }

    printf("\nTotal number of buses :
%d\n",buses);
    return(0); // If you liked this program buy it.
}

```

```
#include <stdio.h>

char input;
long sum=0;

int main()
{
printf("\n\nPlease enter a number : ");
input=getchar();
if (input>=48&&input<59) sum+=(input-48); else // Is the value enter is in [0,9] ?
    {printf("\nInvalid input, Exiting...");return(1);} // Using Ascii values.
while (input=getchar())
    if (input>=48&&input<59) sum+=(input-48); else break;

if (sum%3==0)
    printf("\n\nThe number IS a multiple of 3\n"); else
    printf("\n\nThe number ISN'T a multiple of 3\n");
return 0;
}
```

## HW3-2

// This program approximates the Sinus value given in Radians.  
 // All rights are reserved to Boris Milner & Max Moldavsky.

```

#include<stdio.h>
#include<math.h> // For temporal use only ( Calculating the actual value of Sin(x) )
#define EPSILON 1e-10

double    facto(int base); // Declerations about
double    sinus(int number); // the functions which are
double    mypower(int number,int power); // future to come...
double    myabs(double number);

int    main() // The actual execution of all the fuctions below
{
    double    input,sinvalue,calculated_value;
    printf("\n\nPlease input your number (x) : ");
    if (scanf("%lf",&input)<1)
    {
        printf("\nInvalid data entered, exiting...\n");
        return (1);
    }
    printf("\nThe approximate Sinus of X is : %.8lf",sinvalue=sinus(input));
    printf("\nSame as above, calculated using <math.h> :
%.8lf",calculated_value=sin(input));
    printf("\nApproximate Error is %le",calculated_value-sinvalue);
    return (0);
}

double    sinus(int number) // Calculates the sinus using the given formula
{
    double    value,temp;
    int sign=-1,exponent=3,counter=1;
    value=number;
    do

    {
        temp=(mypower(number,exponent)/facto(exponent))*sign;
        value+=temp;
        exponent+=2;
        sign*=-1;
        counter++;
    }
    while(myabs(temp)>=EPSILON);
    printf("Total number of eterations : %d",counter);
    return (value); // Note - The result is in RADIANS
}

double    facto(int base) // Wrote this to avoid using <math.h>
{
    int    counter;
    double    result=1;

```



## HW3-2

```
for (counter=1;counter<=base;counter++)  
    {  
        result*=counter;  
    }
```

```
return (result);
```

```
}
```

```
double mypower(int number,int power) // Wrote this to avoid using <math.h>
```

```
{  
    double result=1;  
    int counter;  
    for (counter=1;counter<=power;counter++)  
    {  
        result*=number;  
    }
```

```
    return (result);
```

```
}
```

```
double myabs(double number) // Wrote this to avoid using <math.h>
```

```
{  
    if (number<0)  
    {  
        number*=-1;  
    }
```

```
return(number);
```

```
}
```

### HW3-3

```
// This program is a really cool game which randomizes numbers.
// All rights are reserved to Boris Milner & Max Moldavsky.

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MIN -50 // Feel free to put here any MINIMUM value!
#define MAX 50 // Feel free to put here any MAXIMUM value!
#define LIVES 8 // NOTE : MINIMUM MUST NOT BE GREATER THAN THE MAXIMUM !

int guesses,lower,upper,input,answer,options;
int main()
{
    randomize();
    options=(MAX-MIN);
    answer=(MAX-random(options+1)); // Considering random generates from 0 to options-1
    lower=MIN;upper=MAX;
    for (guesses=LIVES;guesses>0;guesses--)

        {printf("\n\nThe number is between %d and %d",lower,upper);
        printf("\nPlease enter your guess (%d) : ",9-guesses);
        if (scanf("%d",&input)!=1) {printf("\nInvalid Input, Exiting...\n"); return(1);}
        if ( (input>upper)||((input<lower) ) {printf("\nInvalid range - You loose !\n"); return(1);}
        if (input<answer) lower=input;
        if (input>answer) upper=input;
        if (input==answer) {printf("\nCongradulations! you win !"); return(0);}
        }
    printf("\nSorry, you are out of guesses - YOU LOST !\n");
    return(0);
}
```

### HW3-4

// This program converts numbers based on 2nd...10th base to decimal value.  
 // All rights are reserved to Boris Milner & Max Moldavsky.

```
#include <stdio.h>
```

```
double    mypower(int number,int power) // Wrote this to avoid using pow <math.h>
        {
            double result=1;
            int counter;
            for (counter=1;counter<=power;counter++)
            {
                result*=number;
            }

            return (result);
        }
```

```
int base,input,counter,limit;
```

```
double dvalue=0;
```

```
int entered[256]; // Because of limitations in variables types
```

```
int main() // definitions the program overflows when using 10th base
```

```
{ // with more than 10 meaningfull digits.
```

```
    // Adjust "entered" value to get more/less meaningfull digits
```

```
    // compability - But be careful : you may cause overflow which
```

```
    // may result in false output.
```

```
do // Here begins the main loop.
```

```
    // Notice : the program assumes the user doesn't DELETE digits.
```

```
    // This assumption is necessary for the program to respont IMEDIATLY.
```

```
{base=input=limit=dvalue=0; // Clears all values from previous calculations.
```

```
for (counter=-1;counter<0;) // Counter is used as a sort of boolean value.
```

```
{
do // Repeats until a valid value is entered.
```

```
{ // Leting the user know the value is invalid (If it's not the 1st etration)...
```

```
printf("\n\nPlease enter a valid base (2..10) : ");
```

```
base=(getche()-48); // getche gets a character from console, and echoes to the screen.
```

```
if (base==53) {printf("\nAdios!\n");return(0);} // If 'e' is entered, program ends.
```

```
limit=(getche()-48); // Temporary use of the variable.
```

```
if (limit==53) {printf("\nAdios!\n");return(0);} // If 'e' is entered, program ends.
```

```
if (limit==0&&base==1) // The user may want the base to be 10.
```

```
{base=10;limit=(getche()-48);
```

```
if (limit==53) {printf("\nAdios!\n");return(0);}
    if (limit!=-2) printf("\nSorry, can't be a valid base\n"); // 10 expected but not found.
```

```
    else counter=0; // 10 expected and indeed 10 found.
```

```
    }
```

```
    else if (limit!=-2) {printf("\nSorry, can't be a valid base\n");} else counter=0;
```

```
}
```

```
while ( base>10 || base<2 ); // Verifies the validity of the base.
```

```
} // Should work for any regular user.
```

```
printf("\n\nPlease enter your number : ");
```

## HW3-4

```
for (counter=0;input!=46&&input!=101;)
    {input=getche();
    if (input==101) {printf("\nAdios!\n");return(0);}
    if (input>=48&&input<=57&&(input-48)<base) {entered[counter]=(input-48);counter++;}
    } // Stores only the meaningfull values in ordered places...
    limit=counter; // From now on, we go backwards and calculate using the given formula...
    while (counter>=0,counter--) dvalue+=(entered[counter]*mypower(base,limit-counter-1));

printf("\n\nThe decimal value is : %.0lf \n",dvalue); // No fraction expected.
}

while( (input-48)==-2 ); // While 'e' is not pressed the program restarts.

return(0);

} // If you liked this program buy it.
```

## Hw4

```
// This program is a clone of a Black Jack game.
// All rights are reserved to Boris Milner & Max Moldavsky.

#include <conio.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

#define STAKE 15
#define PROPABILITY 0.7
#define WINVALUE 21
#define MAXCARD 13
#define MAXTRIES 5

int hitme(double chance); // Propability must be in [0,1]
int user_play(void);
int draw_retries(int retries);

int main()
{
    int user_hand=0,cpu_hand=0,strategy;
    flushall(); // Flushes all open streams.
    clrscr(); // Clears text mode window.
    printf("\n\nPlease Choose CPU's strategy [1,2,3] : ");
    scanf("%d",&strategy);
    if (strategy!=1&&strategy!=2&&strategy!=3)
    {printf("\nIllegal choice, exiting...\n"); exit(1);}
    flushall();
    randomize(); // Macro that initializes random number generator.
    cpu_hand+=(1+rand()%MAXCARD); // CPU draws first card.
    switch (strategy)
    {
        case 1: while (cpu_hand<=STAKE&&cpu_hand<(WINVALUE+1))
        cpu_hand+=(1+rand()%MAXCARD);
        case 2: while (hitme(PROPABILITY)&&cpu_hand<(WINVALUE+1))
        cpu_hand+=(1+rand()%MAXCARD);
        case 3: while
        (hitme(((double)WINVALUE-cpu_hand)/(double)WINVALUE)&&cpu_hand<(WINVALUE+1))
        cpu_hand+=(1+rand()%MAXCARD);
    }

    // &&cpu_hand<(WINVALUE+1) insures that CPU retires as soon as it has overdrew. (As instructed)...

    printf("\nCPU is done...\n\n");
    user_hand=user_play();
    printf("\n\nSeems like you got %d while CPU got %d.\n",user_hand,cpu_hand);
    if (cpu_hand<=WINVALUE && user_hand<=WINVALUE)
        if (cpu_hand<=user_hand)
            if (cpu_hand!=user_hand) printf("\nYou win !!!\n");
            else printf("\nTeko !!!\n");
        else printf("\nCPU wins !!!\n");
    else
    {
        if (cpu_hand<=WINVALUE) printf("\nCPU wins!!!\n");
        if (user_hand<=WINVALUE) printf("\nYou win !!!\n");
        if (user_hand>=WINVALUE && cpu_hand>=WINVALUE) printf("\nTeko !!!\n");
    }
}
```

```

}
return 0;
}

int hitme(double chance) // Propability must be in [0,1]
{
    if (chance>=1) return 1; // Supremum Luck Overflow (SLO)
    if (chance<=0) return 0; // Infimum Luck Overflow (ILO)
    randomize(); // Macro that initializes random number generator.
    return (rand()/(double)chance<=PROPABILITY*chance);
}

int user_play(void)
{
    int counter,process_hand=0,retries;
    char draw_answer;
    printf("\nHow many tries do you want to get on each draw? (1..%d) : ",MAXTRIES);
    scanf("%d",&retries);
    if (retries>MAXTRIES || retries<1) {printf("\nIllegal configuration, exiting...\n"); exit(1);}
    flushall();
    while (draw_answer!='n')
        {
            if (process_hand>WINVALUE) break;
            printf("\n\nYour Score so far is : %d",process_hand);
            printf("\nDo you want to draw a card? : ");
            scanf("%c",&draw_answer);
            flushall();
            if (draw_answer=='n') break;
            else if (draw_answer!='y') {printf("\nIllegal answer, exiting...\n"); exit(1);}
            randomize();
            process_hand+=(draw_retries(retries));
            if (process_hand>WINVALUE) draw_answer='n';
        }

    return(process_hand);
}

int draw_retries(int retries)
{
    int temp_hand;
    char answer;
    for ( ; retries>0 ; retries--)
        {
            printf("\nYou draw : %d",temp_hand=(1+rand()%MAXCARD) );
            printf("\nYou got a total of %d tries left.",retries-1);
            if (retries>1) {printf(" --> redraw? (y/n) : ");
            scanf("%c",&answer);}
            flushall();
            if (answer=='n') retries=0;
            else if (answer!='y'&&retries>1) {printf("\nInput Error, exiting...\n"); exit(1);}
        }
    return temp_hand;
}

```

## HW5-1

```
// This program demonstrates work with files - input based calculations.
// All rights are reserved to Boris Milner & Max Moldavsky.

#include <stdio.h>
#include <conio.h>

#define MAX_FILE_NAME_LENGTH 8
#define FILE_EXTENTION ".txt"
#define FILE_EXTENTION_LENGTH 4

int main()
{
    FILE *in, *out;
    char input[12],output[12];//[0,7]=filename::8='.'::[9,11]=postfix::12='\n'
    char op; // States the operation to be done.
    double first,second; // The first and the second operands.
    int counter;

    clrscr(); // Clears text mode window.
    input[MAX_FILE_NAME_LENGTH]=output[MAX_FILE_NAME_LENGTH]=NULL;
    printf("\nPlease enter a valid INPUT filename without the postfix : ");
    if (scanf("%s",&input)!=1) {printf("\nFilename error, program terminated.\n");return 1;}
    if (input[MAX_FILE_NAME_LENGTH]!=NULL) {printf("\nFilename error, program
terminated.\n");return 1;}
    strcat(input,FILE_EXTENTION); // Appends one string to another.
    if ((in = fopen(input, "r+"))== NULL)
    {fprintf(stderr, "\nCannot open input file, exiting...\n");return 1;}
    else printf("\n%s was succesfully found.\n",input);
    printf("\nPlease enter a valid OUTPUT filename without the postfix : ");
    if (scanf("%s",&output)!=1) {printf("\nFilename error, program terminated.\n");return 1;}
    if (output[MAX_FILE_NAME_LENGTH]!=NULL) {printf("\nFilename error, program
terminated.\n");return 1;}
    strcat(output,FILE_EXTENTION); // Appends one string to another.
    if ((out = fopen(output, "wt"))== NULL)
    {fprintf(stderr, "\nCannot open output file, exiting...\n");return 1;}
    else printf("\n%s was succesfully created.\n",output);

    fscanf(in,"%d\n",&counter); // Expects information as instructed.

    for ( ; counter>0; counter--)
    {
        fscanf(in,"%c\n:\n%lf\n%lf\n",&op,&first,&second);
        if (op=='a' || op=='A')
            if (fprintf(out,"%lf\n",first+second)==0)
                {fprintf(stderr, "\nUnexpected error writing to %s !\n\n",output);exit(1);}
        if (op=='m' || op=='M')
            if (fprintf(out,"%lf\n",first*second)==0)
                {fprintf(stderr, "\nUnexpected error writing to %s !\n\n",output);exit(1);}
        if ( (op=='d' || op=='D') && second!=0.0 )
            if (fprintf(out,"%lf\n",first/second)==0)
                {fprintf(stderr, "\nUnexpected error writing to %s !\n\n",output);exit(1);}
        if ( (op=='d' || op=='D') && second==0.0 )
            if (fprintf(out,"Division by zero.\n")==0)
                {fprintf(stderr, "\nUnexpected error writing to %s !\n\n",output);exit(1);}

    if ( op!='a' && op!='A' && op!='m' && op!='M' && op!='d' && op!='D' )

```

## HW5-1

```
{fprintf(stderr, "Unexpected input, exiting...\n");exit(1);}  
}  
fclose(in); // Closes a stream.  
fclose(out); // Closes a stream.  
  
printf("\nSeems like evrything went smoooth like butter!\n");  
return 0;  
}
```



## Hw5-2

```
// This program reports wether a string contains each letter of the english ABC.  
// All rights are reserved to Boris Milner & Max Moldavsky.
```

```
#include <stdio.h>  
#include <conio.h>
```

```
int string_contains_alphabet(char *string);
```

```
int main() {  
    int res; // Following is the tested string array.  
    char str[]="The_Quick_Brown_Fox-Jumps_Over_The_Lazy_Dog.";  
    res = string_contains_alphabet(str);  
    clrscr(); // Clears text mode window.  
    printf("The string %s the entire alphabet!\n",  
        res ? "contains" : "does not contain");  
    return 0;  
}
```

```
int string_contains_alphabet(char *string)  
{  
    int counter=0;  
    char alphabet[25]; // 26 letters in the alphabet.  
  
    for (counter=0;counter<26;counter++) alphabet[counter]=-1;  
    for (counter=0;string[counter]!=NULL;counter++)  
    {  
        if (string[counter]>='A'&&string[counter]<='z')  
            if (string[counter]>='a') alphabet[string[counter]-'a']=(string[counter]-'a');  
            else alphabet[string[counter]-'A']=(string[counter]-'A');  
    }  
    for (counter=0;counter<26;counter++)  
        if (alphabet[counter]!=counter) return 0;  
  
    return 1;  
}
```

### Hw5-3

```
// This program sorts an array of integers [1st even then odds].  
// All rights are reserved to Boris Milner & Max Moldavsky.
```

```
#include <stdio.h>
```

```
void sort_even_odd(int array[], int n);
```

```
int main()
```

```
{ // Following is the array to sort.
```

```
int array[] = {92,5,7,4,5,65,80,643,23,2,45,2,3,1};
```

```
int i, n=sizeof(array)/sizeof(int); // n contains the size of the array.
```

```
    printf("\n\nThe original array was : ");  
    for (i=0; i<n; i++) printf("%d ",array[i]);  
    printf("\nWe sorted it and it became : ");
```

```
sort_even_odd(array,n);
```

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

```
printf("%d ",array[i]);
```

```
printf("\n");
```

```
return 0;
```

```
}
```

```
void sort_even_odd(int array[], int n)
```

```
{
```

```
int counter,i,temp;
```

```
for (counter=n;counter>0;counter--)
```

```
for (i=0;i<counter-1;i++)
```

```
if (array[i+1]<array[i])
```

```
{ temp=array[i+1];array[i+1]=array[i];array[i]=temp; }
```

```
for (counter=n;counter>0;counter--)
```

```
for (i=0;i<counter-1;i++)
```

```
if (array[i+1]%2==0&&array[i]%2!=0)
```

```
{ temp=array[i+1];array[i+1]=array[i];array[i]=temp; }
```

```
return;
```

```
}
```

## Hw5-4

// This main() was written for testing purposes only, you may ignore it.  
 // All rights are reserved to Boris Milner & Max Moldavsky.

```

/*-----*/
//
#include <stdio.h> //

//
void set_union(char s1[],char s2[], int n, int m, char s3[], int *size); //
void set_intersect(char s1[],char s2[], int n, int m, char s3[], int *size); //

//
int main() //
{ //

//
char str1[]={'A','b','b'},str2[]={'A','H','H','a','b'},str3[6]; //

//
int c; //
int *bignes; //

//
set_union(str1,str2,2,4,str3,bignes); //
printf("\n\nThe Union is : ");for (c=0;c<*bignes;c++)printf("%c",str3[c]); //
printf("\n\nThe Size of outcome is : %d", *bignes); //

//
set_intersect(str1,str2,2,4,str3,bignes); //
printf("\n\nThe Intersect is : ");for (c=0;c<*bignes;c++)printf("%c",str3[c]); //
printf("\n\nThe Size of outcome is : %d", *bignes); //
return 0; //
} //
/*-----*/

```

// The following functions calculate UNION & INTERSECT of 2 sorter arrays.  
 // All rights are reserved to Boris Milner & Max Moldavsky.

```

void set_union(char s1[],char s2[], int n, int m, char s3[], int *size)
{
/* n --> size of s1 <==> m--> size of s2 --> Note than arrays begin with 0... */

/* cnt1 --> counter for s1 <==> cnt2 --> counter for s2 */

int cnt1,cnt2,counter;
cnt1=cnt2=counter=0;

while (cnt1!=(n+1)&&cnt2!=(m+1)) // While not end of array...
{
if (s1[cnt1]>s2[cnt2])
{

{
while (s1[cnt1]==s1[cnt1+1]) cnt1++; // Skips clones...
while (s2[cnt2]==s2[cnt2+1]) cnt2++; // Skips clones...

```

```

    }

    s3[counter]=s2[cnt2];
    cnt2++;counter++;

}

if (s2[cnt2]>s1[cnt1])
{

{
while (s1[cnt1]==s1[cnt1+1]) cnt1++; // Skips clones...
while (s2[cnt2]==s2[cnt2+1]) cnt2++; // Skips clones...
}

s3[counter]=s1[cnt1];cnt1++;counter++;

}

if (s2[cnt2]==s1[cnt1])
{

{
while (s1[cnt1]==s1[cnt1+1]) cnt1++;
while (s2[cnt2]==s2[cnt2+1]) cnt2++;
}

s3[counter]=s1[cnt1];cnt1++;
cnt2++;counter++;

}

}

if (cnt1==(n+1))
while (cnt2!=(m+1)) {s3[counter]=s2[cnt2];cnt2++;counter++;}
if (cnt2==(m+1))
while (cnt1!=(n+1)) {s3[counter]=s1[cnt1];cnt1++;counter++;}
*size=counter;
return; // Void functions may not return a value.
}

void set_intersect(char s1[],char s2[], int n, int m, char s3[], int *size)
{
int cnt1,cnt2,counter;
cnt1=cnt2=counter=0;

while (cnt1!=(n+1)&&cnt2!=(m+1))
{
if (s2[cnt2]==s1[cnt1]) {s3[counter]=s1[cnt1];cnt1++;cnt2++;counter++;}
else s1[cnt1]>=s2[cnt2]? cnt2++ : cnt1++;
}
*size=counter;
return; // Void functions may not return a value.
}

```

## HW6-1

```
// This program finds the 1st longest sequence of digits in a given array.  
// All rights are reserved to Boris Milner & Max Moldavsky.
```

```
#include <stdio.h>
```

```
void print_seq(int a[], int n);
```

```
int main()
```

```
{  
int array[] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14};
```

```
clrscr(); // Clears text mode window.
```

```
print_seq(array,13);
```

```
return 0;
```

```
}
```

```
/*+++++*/
```

```
void print_seq(int a[], int n)
```

```
{  
int value,repeats,temp=1,counter=0;
```

```
value=a[0];
```

```
repeats=1;
```

```
while (counter<n) // Assumes there IS ONE longest sequence...
```

```
{
```

```
while (a[counter+1]-a[counter]==1&&counter+1!=(n+1)) {temp++;counter++;}
```

```
if (temp>repeats) {repeats=temp;value=a[counter];} // Complexity --> O(n)
```

```
temp=1;
```

```
counter++;
```

```
}
```

```
if (temp==repeats==1) printf("\nSorry, no sequence found.\n"); else
```

```
printf("\n\nLongest sequence is of %d, it appeared : %d times !",value,repeats);
```

```
return;
```

```
}
```

## HW6-2

```
// This program detects whether a number divides by 7 using a given formula.  
// All rights are reserved to Boris Milner & Max Moldavsky.
```

```
/* Answer to section B
```

```
-----
```

```
n<=0 ----> result = 1
```

```
n>=1 ----> result = (2^n)*(Sigma of n {1 to n})
```

```
n is not an integer ----> compilation error */
```

```
#include <stdio.h>
```

```
int main()
```

```
{long int input; // To increase compability use UNSIGNED long instead  
clrscr(); // Clears text mode window.  
printf("\n\nPlease enter input : "); // and input only ABS value of the number.  
scanf("%ld",&input);  
if (input<0) input*=-1; // Absolut value of input.  
while (input>10)  
{  
if (input/10<(input%10)+(input%10) )  
{printf("\n\nDoesn't divide by 7 !!");return 0;}  
  
input=(0 - ( (input%10)+(input%10) ) + input/10 );  
}  
if ((input==0)||input==7)  
printf("\n\nDOES divide by 7 !!\n"); else printf("\n\nDOESN'T divide by 7 !!\n");  
  
return 0;  
}
```

```

#include <stdio.h>

int find_x (int a);
int find_y (int a, int y); // Must be referred with 0 as the initial value of y.
int find_z (int a);
int compare(int a,int b);
void sort(int a[],int n);
void merge(int a[],int na, int b[],int nb, int c[]);
void mergesort(int a[],int n, int temp[]);

int main()
{
int a[5]={160,80,16,24,45};
int i,tmp[5];
mergesort(a,5,tmp) ;
for(i=0; i<5; i++)
{
printf(" %d ",a[i]);
}
return 0;
}

int find_y (int a, int y) // Must be referred with 0 as the initial value of y.
{
if (a%2!=0) return y;
else return ( find_y(a/2,++y) );
}

int find_z (int a)
{
int z;
z=0;
while (a%3==0) {a/=3;z++;}
return z;
}

int find_x (int a)
{
while (a%3==0) a/=3;
while (a%2==0) a/=2;
return a;
}

int compare(int a,int b)
{
int x1,x2,y1,y2,z1,z2;
x1=find_x(a); x2=find_x(b);
y1=find_y(a,0); y2=find_y(b,0);
z1=find_z(a); z2=find_z(b);

if ( (x1<x2)||((x1==x2)&&(y1<y2)) || ( (x1==x2)&&(y1==y2)&&(z1<z2) ) )
return 0;
else return 1;
}

void merge(int a[],int na, int b[],int nb, int c[])

```

```

{
int ia=0,ib=0,ic=0;
while ( (ia<na)&&(ib<nb) )
if ( compare(a[ia],b[ib]) ) c[ic++]=a[ia++];
else c[ic++]=b[ib++];
while (ia<na) c[ic++]=a[ia++];
while (ib<nb) c[ic++]=b[ib++];
}

```

```

void mergesort(int a[],int n, int tmp[])

```

```

{
int counter;
int m=n/2;
if (n<=1) return;
mergesort(a,m,tmp); // Left half.
mergesort(a+m,n-m,tmp); // Right half.
merge(a,m,a+m,n-m,tmp);
for (counter=0;counter<n;counter++) a[counter]=tmp[counter];
}

```

```

/*
void sort(int a[],int n)

```

```

{
int *tmp;
tmp=calloc(n,sizeof(int));
mergesore(a,n,tmp);
free(tmp);
} */

```