***Some String Functions:***

***strlen(s1)*** Returns length of the string s1

***strcat( s1 , s2 )*** Concatenates a copy of string s2 onto the string s1

***strncat(s1, s2, n)*** Concatenates a copy of up to n characters from string s2 onto the string

in s1

**strcpy(s1, s2)** Copies the string s2 to s1

**strncpy(s1, s2, n)** Copies a string up to **n** characters from s2 to s1

***strcmp(s1,s2)*** Compares the string s2 with the string s1. Returns an integer less than, equal

to, or greater than depending on the result of the comparison

***strncmp(s1, s2, n)*** Compares at most the first n characters of the string s1 to s2, and

returns an integer less than, equal to , or greater than depending on the result of the

comparison

***strchr(s1 , c) L***ocates the first occurrence of c(a char) in the string s1, and returns a pointer

to the located character. If the search is not successful than null is returned

***strrchr(s1, c) L***ocates the last occurrence of c(a char) in the string s1, and returns a pointer

to the located character. If the search is not successful than null is returned

**Q1) (48 pts)** Give the outputs for each of the following C program in the corresponding boxes on the right:

**4**

**9**

**13**

**a)**

int main()

{

 int array[] = {1,2,4,5,4,0,3};

 int \*p, sum=0;

 p = array+1;

 while(\*++p)

 {

 sum += \*p;

 printf("%d\n",sum);

 }

 return 0;

}

**-----------------------------------------------------------**

**b)**

1. **1**
2. **1**
3. **1**

#include<stdio.h>

int f(int i, int \*b)

{

 i = i + 1;

 if (\*b-1)

 return \*++b;

 else

 return \*b++;

}

main()

{

 int i, a[]={1, 2, 1};

 for(i=0; i<3; i++)

 printf("%d %d\n", i, f(i, a+i));

 return 0;

}

 **-----------------------------------------------------------**

**c)**

#include<stdio.h>

**12**

**13**

**14**

**35**

**6**

**0**

**0**

#define SIZE 7

void fun1(int [ ], int);

int main()

{ int i;

int array1[SIZE] = {2, 3, 4, 5, 6};

fun1(array1, array1[2]);

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

 printf(“%\nd”, array1[i]);

return 0;

}

void fun1(int array1[], int len)

{ int i;

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

 array1[i] += 10;

array1[len-1] += 20;

}

**-----------------------------------------------------------**

**d)**

#include <stdio.h>

 **a = 9 b = 6**

void fun(int \*p, int x)

{

 \*p += 5;

 x += 5;

}

main()

{

 int a = 4, b = 6;

 fun(&a, b);

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

}

**------------------------------------------------------------------------**

**e)**

**18 11 7 10 9**

#include <stdio.h>

main()

{

 int k, \*p, v[] = {2, 5, 3, 4, 6};

 p = v + 2;

 \*p = 7;

 p[2] = 9;

 p[-1] = 11;

 v[0] = p[1] + 14;

 \*(v+3) = \*p + 3;

 for(k = 0; k <= 4; k++) printf("%d ", v[k]);

}

**-------------------------------------------------------------------------**

**f)**

#include <stdio.h>

main()

**12
12
 4 3**

{

 int x, \*p, v[] = {2, 5, 3, 4, 6};

 p = v;

 x = ++\*p + 9;

 printf("%d\n", x);

 x = \*p++ + 9;

 printf("%d\n", x);

 printf("%d %d\n", p[2], v[0]);

**}**

**-------------------------------------------------------------------------**

**g)**

#include <stdio.h>

**x[0] = 0
x[1] = 10
x[2] = 4
x[3] = 2
x[4] = 1**

main()

{

 int x[5] = {0};

 int j, k;

 for(j=1; j < 5; j++)

 for(k=1; k < 5; k++) x[k] += j/k;

 for(j=0; j < 5; j++)

 printf("x[%d] = %d\n", j, x[j]);

}

------------------------------------------------------------------------------------------------------------------

**h)** #include <stdio.h>

**2 5 13 15 18 9**

void fun(int \*p, int size)

{

 int k;

 for(k=0; k<size; k++) p[k] += k+10;

}

main()

{

 int k, v[] = {2, 5, 3, 4, 6, 9};

 fun(v+2, 3);

 for(k=0; k<= 5; k++) printf("%d ", v[k]);

**}**

**Q2) (10 pts)** Give the output of the following program in the corresponding answer box.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

main()

{

 char s1[20]="ZERO", s2[20]="ONE", s3[20]="TWO";

 printf("%d\n", strlen(s1));

 strcat(s2,s3);

**4
ONETWO
ZETW**

 printf("%s\n",s2);

 strncpy(s1+2,s3,2);

 printf("%s\n",s1);

}

**Q3)** **(12 pts)** The following C program computes the sum of the squares of the digits of a given positive integer. Fill in the blanks such that the program works accordingly.

*(For example, if the positive integer is 345, then the function will compute 3\*3 + 4\*4 + 5\*5 = 50* )

#include<stdio.h>

int main()

{

int sum=0, num, digit;

/\* num is the number entered, digit is the respective digit \*/

printf(“Enter a number:”);

scanf(“%d”,&num);

do

{

digit=\_\_\_\_\_num%10\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

sum+=\_\_\_\_\_digit\*digit\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

num=\_\_\_\_\_\_num/10\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

} while(\_\_\_\_\_num!=0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

printf(“Sum of the squares of digits = %d”,sum);

return 0;

}

**Q4)** **(12 pts)** The following programchecks the contents of a two dimensional integer array, **A**, and counts the number of rows having all elements with zero values. It uses an integer variable *flag* which is assigned to zero if the corresponding array element is non-zero, and assigned to 1 if the array element is 1. Assume that the array **A** contains 30 rows and 20 columns with integer elements. Fill in the blanks such that the program works accordingly.

#include <stdio.h>

Int main ()

{

 int A [30][20];

 int r, c, flag, sum=0; /\* sum is the number of all-zero rows \*/

 for(r=0 ;r<30 ; r++)

 for(c=0; c<20; c++)

 scanf(“%d”, &A[r][c]); /\*enter integer elements for the array \*/

 for(r=0 ;r<30 ; r++)

 {

flag=\_0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 for(c=0; c<20; c++)

 {

 if (\_\_\_A[r][c]!=0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

flag=\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 }

if (\_\_\_flag==0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

 sum++;

 }

 return 0;

 }

**Q5) (18 pts)** Consider the following C program.

Function **fun1( )** calculates the largest element of the array **A**and stores its value in the **A[0]** position. The largest value is displayed in the main program The built-in function *rand( )* is used to generate non-negative integers randomly.

Write the prototype of the function **fun1( )**, call it in the main program and complete its definition by filling in the blanks.

#include<stdio.h>

#include<stdlib.h>

­­­­­­­­­­­­­­­­­­­­­­­­­\_void fun1 (int A[ ], int j\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; /\* Write the prototype of fun1( ) \*/

int main(void)

{

 int A[10],i

 for( i=0; i<10; i++)

 A[i] = rand( )%9;

\_\_\_fun1( A, 10 )\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ; /\* call function fun1( ) \*/

printf("The largest value of array is %d\n", A[0]);

 return 0;

}

void fun1(\_\_int B[ ], int j \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) /\* definition of function fun1( ) \*/

{

int temp = B[0],i; /\* temp stores the largest element \*/

for(i=1; i<j; i++)

 if(\_\_temp < B[i]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

\_\_\_\_\_ temp = B[i]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

B[0] = \_temp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 return;

}

**Q6) (15 pts)** The program below reads a text file named clients.txt that contains the following 5 lines of client information including the account number, name, and balance:

100 Jones 24.98

200 Doe 345.67

300 White 0.00

400 Stone -42.16

500 Rich 224.62

The program prints the data read from the text file to the screen. Complete the missing parts of the program (there are 5 blanks).

#include <stdio.h>

int main() {

    \_\_\_FILE \*fp;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // Declare a file pointer

    int account; // account number

    char name[30]; // account name

    double balance; // account balance

    // Open the file for reading

 \_\_\_fp=fopen("clients.txt","r");\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 // Check if the file has been opened successfully

    if (\_\_fp==NULL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) {

        puts("File could not be opened");

    }

    else {

        printf("%-10s%-13s%s\n", "Account", "Name", "Balance");

        for (int i=0; i<5; i++) {

 // read account, name and balance from file

            \_\_fscanf(fp, "%d%s%lf", &account, name, &balance);\_\_

            printf("%-10d%-13s%7.2f\n", account, name, balance);

        }

        \_\_\_fclose(fp);\_\_\_\_\_\_\_\_\_\_\_\_\_ // Close the file

    }

    return 0;

}