Fast**National University of Computer & Emerging Sciences, Karachi  
Fall-2018Computer Science Department  
Final Exam  
17th December 2018, 09:00 am – 12:00 (noon)**

|  |  |  |
| --- | --- | --- |
| **Course Code: CS118** | **Course Name: Programming Fundamentals** | |
| **Instructor Name: M. Shahzad / Dr. Farooque / Shoaib Rauf / Tania Iram** | | |
| **Student Roll No:** | | **Section No:** |

**SOLUTION PAPER**

Instructions:

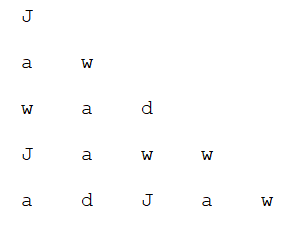
* Return the question paper and make sure to keep it inside your answer sheet.
* Read each question completely before answering it. There are **8 questions and 3 page**.
* In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
* You are **not allowed to write** anything on the question paper (except your ID and group).

**Time**: 180 minutes. **Max Points**: 53 Points

**Question 1:** Observe and try to understand the following programs. Write errors if there are any available or write outputs if the programs are fine. **[5 points]**

|  |  |
| --- | --- |
| (i)  int main()  {  char \*s1 = (char \*)malloc(50);  char \*s2 = (char \*)malloc(50);  strcpy(s1, "Hello");  strcpy(s2, "World");  strcat(s1, s2);  printf("%s", s1);  return 0;  }  **Ans:**  **Helloworld** | (ii)  void main()  {  int k=5;  int \*p=&k;  int \*\*m=&p;  printf("%d %d %d",k,\*p,\*\*m);  }  **Ans:**  **5 5 5** |
| (iii)  int main()  {  int arri[] = {1, 2 ,3};  int \*ptri = arri;  char arrc[] = {1, 2 ,3};  char \*ptrc = arrc;  printf("sizeof arri[] = %d ", sizeof(arri));  printf("sizeof ptri = %d ", sizeof(ptri));  printf("sizeof arrc[] = %d ", sizeof(arrc));  printf("sizeof ptrc = %d ", sizeof(ptrc));  return 0;  }  **Ans:**  **sizeof arri[] = 12 sizeof ptri = 4 sizeof arrc[] = 3 sizeof ptrc = 4** | (iv)  int main()  {  int i = 0;  for (i=0; i<20; i++)  {  switch(i)  {  case 0:  i += 5;  case 1:  i += 2;  case 5:  i += 5;  default:  i += 4;  break;  }  printf("%d ", i);  } return 0;  }  **Ans:**  **16 20** |
| (v)  int main()  {  int a = 12;  void \*ptr = &a;  printf("%d", \*(int \*)ptr);  getchar();  return 0;  }  **Ans:**  **12** |  |

**Question 2:** Print the following output using a C program. Take input name and print as triangle shape using each character of the name, ex. Input= “Jawwad”. **[6 points]**



**SOLUTION:**

#include<stdio.h>

#include<string.h>

int main()

{

int i,j;

char a[20];

int b=0;

gets(a);

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

{

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

{

printf("%5c",a[b]);

b++;

if(b==strlen(a))

b=0;

}

printf("\n\n");

}

return 0;

}

**Question 3:** Sajid wants to perform operation on a file. Help him write a program to count the number of rows stored in a file (.txt). What file mode will be a better choice for him and why?

**[6 points]**

**SOLUTION:**

#include <stdio.h>

int main()

{

FILE \*fp;

int no\_lines = 0;

char filename[40], sample\_chr;

printf("Enter file name: ");

scanf("%s", filename);

fp = fopen(filename, "r");

sample\_chr = getc(fp);

while (sample\_chr != EOF) {

if (sample\_chr == '')

{

no\_lines=no\_lines+1;

}

sample\_chr = getc(fp);

}

fclose(fp);

printf("There are %d lines in %s ", no\_lines, filename);

return 0;

}

File mode “r” would be a simpler and better choice because he can retrieve and count the lines.

**Question 4:** Create three text files named as Department.txt, Personal.txt and Combine.txt.

Personal file contains ID and Name, Department file contains ID and Salary. Write a function which takes input as record IDs and gets the detail from both personal and department file and then adds this entry into combine file (ID, Name, Salary).

**[6 points]**

**SOLUTION:**

#include<stdio.h>

#include<stdlib.h>

void add(int);

struct personal{

int id;

char name[20];

}p[3] = {{1, "Asad",}, {2, "Bilal"}, {3, "Imran"}};

struct department{

int id;

float salary;

}d[3]={{1, 50000.0}, {2, 25000.0}, {3, 30000.0}};

int main()

{

int id;

char ch='y';

FILE \*dpt = fopen("Department.txt", "a+");

fwrite(d,sizeof(d),1, dpt);

fclose(dpt);

FILE \*per = fopen("Personal.txt", "a+");

fwrite(p,sizeof(p),1, per);

fclose(per);

printf("Enter Record ID: ");

scanf("%d", &id);

add(id);

while(ch!='n'){

printf("Record Combined...\n");

printf("Do you want to combine again(y/n): ");

scanf(" %c",&ch);

if(ch=='y'){

printf("\nEnter Record ID: ");

scanf("%d", &id);

add(id);

}

}

}

void add(int id){

FILE \*cb = fopen("Combine.txt", "a+");

FILE \*dpt = fopen("Department.txt", "r");

FILE \*per = fopen("Personal.txt", "r");

struct personal pRead;

struct department dRead;

while(!feof(per)){

fread(&pRead,sizeof(pRead),1, per);

if(pRead.id==id){

while(!feof(dpt)){

fread(&dRead, sizeof(dRead), 1, dpt);

if(dRead.id==id){

fprintf(cb,"%d %s %f\n", pRead.id,pRead.name,dRead.salary);

fclose(dpt);

break;

}

}

fclose(per);

break;

}

}

}

**Question 5:** Ali needs to compile result of two section together. Develop a system to merge the data from 2 different size arrays in 1 array by passing to a function using pointers. Also, return the address of new array and print this new Array from Main Function. **[6 points]**

*void\* MergeArray (const void \*Array1, size\_t size1, const void \*Array2, size\_t size2);*

Hint: Don’t use any built-in function. Use dynamic memory allocation.

**SOLUTION:**

#include<stdio.h>

#include<stdlib.h>

void\* MergeArray(const void \*Aray1,size\_t size1,const void \*Aray2,size\_t size2)

{

char \*ptr;

int i,j,k;

int x=size1+size2;

char \*Array1 = (char \*) Aray1;

char \*Array2 = (char \*) Aray2;

ptr=(char \*)calloc(x,sizeof(char));

for(i=0;i<size1;i++)

{

ptr[i]= \*(Array1 + i);

}

k=0;

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

{

ptr[j]=\*(Array2 + k++);

}

return ptr;

}

void main()

{

int one,two;

int i=0;

char \*a;

printf("Enter size of 1st array: ");

scanf("%d",&one);

printf("Enter size of 2nd array: ");

scanf("%d",&two);

char a1[one],a2[two];

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

{

printf("\nEnter Elements at %d index of Array1",i);

scanf(" %c",&a1[i]);

}

printf("\n\n");

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

{

printf("\nEnter Elements at %d index of Array2",i);

scanf(" %c",&a2[i]);

}

a=(char \*)MergeArray(&a1,one,&a2,two);

int b=one+two;

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

{

printf("%c",\*(a+i));

}

}

**Question 6:** Develop a system for a queue management for a exhibition ticketing service, for a maximum of 50 people. Each person in queue has a ticket number and name (Hint: Use Structures). A queue is a first in first out data store technique. Write four functions as follows:

**[12 points = 3 + 3 + 3 + 3]**

1. A function which inserts new person in the queue.
2. A function which removes a person from queue.
3. A function to selects a person on the basis of given name. Print the data using pointer to structures.
4. A function which initializes a pointer to function, for each of above functions and calls using these new pointers. (Hint: Signature of functions must be same)

|  |
| --- |
| #include<stdio.h>  #include<string.h>  typedef struct{  int num;  char name[50];  } person;  void main()  {  int count=0;  person q[50];  pointers(q,&count);  } |
| (a)  void insert(person q[], int \*count)  {  scanf("%d",&q[\*count].num);  scanf("%s",q[\*count].name);  (\*count)++;  } |
| (b)  void rem(person q[], int \*count)  {  if(\*count>0)  {  for(int a=0;a<\*count;a++)  q[a]=q[a+1];  (\*count)--;  }  } |
| (c)  void select(person q[], int \*count)  {  person \*ptr;  int t=0;  char temp[50];  scanf("%s",temp);  for(int a=0;a<\*count;a++)  if(strcmp(temp,q[a].name)==0)  {  t=a;  break;  }  ptr=&q[t];  printf("%d and %s",ptr->num, ptr->name);  } |
| (d)  void pointers(person q[], int \*count)  {  void (\*ptr1)(person \*, int \*);  ptr1=rem;  void (\*ptr2)(person \*, int \*);  ptr1=select;  void (\*ptr3)(person \*, int \*);  ptr1=insert;  ptr3(q,count);  ptr1(q,count);  ptr2(q,count);  } |

**Question 7:** Write a program which inputs inventory information from the user. Inventory information includes paper\_order, ribbon\_order and ink\_order amounts. The program also asks user for an input as task\_value (character) to select an operation based on the value of inventory.

**[6 points]**

* Increment total\_paper by paper\_order if task\_value is 'B' or 'C’;
* increment total\_ribbon by ribbon\_order if task\_value is 'E’, 'F’, or 'D’.
* Increment total\_ink by ink\_order if task\_value is 'A' or 'X’.
* If task\_value is 'M' then print total\_paper, total\_ribbon and total\_ink.
* Display an error message if the value of task\_value is not one of these eight letters.

(Note: the values of total\_paper, total\_ribbon and total\_ink are already declared in the program.)

**SOLUTION:**

|  |
| --- |
| Include<stdio.h>  intmain() |
| { int pap\_order,ribbon\_order, ink\_order;  Float total\_paper, total\_ribbon, total\_ink;  Char inv\_value;  total\_paper= total\_ribbon=total\_ink=0;  printf(“enter ‘B or C’ for paper order\n, ‘E,F,D’ for ribbon order\n, A or X for ink\_order\n, enter M to print previous order values:\n”);  scanf(“%c”,&inv\_value);  switch(inv\_value){  case ‘B’:  case ‘C’:  printf(“Enter the new order quantity for paper”);  scanf(“%f”,&paper\_order);  total\_paper+=paper\_order;  break;  case ‘E’:  case ‘F’:  case ‘D’:  printf(“Enter the new order quantity for ribbon”);  scanf(“%f”,&ribbon\_order);  total\_ribbon+=ribbon\_order;  break;  case ‘A’:  case ‘X’:  printf(“Enter the new order quantity for ink in ML”);  scanf(“%f”,&ink\_order);  total\_ink+=ink\_order;  break;  case ‘M’:  printf(“Paper\_order:%f\nRibbon\_order:%f\nInk\_order:%f”, total\_paper, total\_ribbon, total\_ink);  break;  default:  printf(“Wrong character entered”);  break;  } return 0;} |

**Question 8:** A junkyard wants to keep track of how much tons of junk each of its three junk trucks collect each day during a typical week. Write a program that stores this information in a two dimensional 3 × 7 array, where each row represents a different junk truck and each column represents a different day of the week. The program should first have the user input the data for each junk truck. Then it should create a report that includes the following information:  **[6 points]**

* Average quantity of junk collected per day by all the trucks.
* The least amount of junk collected during the week by any one truck.
* The greatest amount of junk collected during the week by any one truck.

**SOLUTION:**

|  |
| --- |
| Include<stdio.h>  intmain() |
| { |
| // Create a two-dimensional 3x7 array. |
| constint junk\_t = 3; |
| constint days = 7; |
| int junk[junk\_t][days]; |
| double tons, least, most, sum = 0; |
|  |
|  |
| // Ask user to input data for each junk\_truck |
| Printf( "Input how many tons of junk each junk\_truck collected each day.\n"; |
| for (int row = 0; row < junk\_t; row++) |
| { |
| for (int col = 0; col < days; col++) |
| { |
| do |
| { |
|  |
| Printf( "Junk\_truck # %d on day #%d collected%d:”, (row + 1), (col + 1)); |
|  |
| Scanf(“%f”,&tons); |
|  |
| if (tons <0) |
| { |
| Printf( "Error! Number of tons must be greater than 0.\n"); |
| } |
|  |
| } while (tons <0); |
|  |
| junk[row][col] = tons; |
|  |
|  |
| // Get total junk collected per day by the whole convoy of junk\_trucks. |
| sum += tons; |
| } |
| printf(“\n”); // make blank space |
| } |
|  |
| // Get least and greatest amount of junk collected by any one junk\_truck. |
| least = most = junk[0][0]; |
| for (int row = 0; row < junk\_t; row++) |
| { |
| for (int col = 0; col < days; col++) |
| { |
|  |
| if (junk[row][col] < least) |
| { |
| least = junk[row][col]; |
| } |
| if (junk[row][col] > most) |
| { |
| most = junk[row][col]; |
| } |
| } |
| } |
|  |
| printf( "\n Junk\_truck Weekly Junk Report\n" |
| by the whole convoy of junk\_trucks\n" |
| "----------------------------------------------------\n"); |
|  |
| printf( "Average amount of junk collected per day is %F tons: " |
| ,sum / 21.0 ); |
| printf( "Least amount of junk ecollected in tons: %d\n",tons); |
| printf( "Greatest amount of junk ecollected in tons:%d\n",most); |
|  |
| return0; |
| } |

***BEST OF LUCK!***