**Sample Questions for CT2**

1. Write a function that receives 3 arguments – an array of numbers, 2 variables – small and large.
2. The function finds the largest and the smallest element in the array and stores them in the corresponding variables passed as arguments.
3. Call this function from the main program by passing all the arguments by pointer and print the result in the main program.

Do not return any values from the function back to the calling program.

**Solution:**

| #include<stdio.h> void minmax(int arr[], int\* maxcumszp, int\* minp){  int sz = \*maxcumszp;  int min, max;  min = max = arr[0];  for(int i = 1; i < sz; i++){  if(arr[i] > max)  max = arr[i];  if(arr[i] < min)  min = arr[i];  }  \*maxcumszp = max;  \*minp = min; }  int main(){  int a[10];  for(int i = 0; i < 10; i++)  scanf("%d", &a[i]);  int maxcumsz, min;  maxcumsz = 10;  minmax(a, &maxcumsz, &min);  printf("Max = %d, Min = %d", maxcumsz, min);  return(0); } |
| --- |
| The above program passes the size of the array into the variable ‘maxcumszp’ while calling the function, and puts the maximum value into this variable before returning. |

2. Write a function to calculate parking charges of a vehicle. Enter the type of vehicle as character (c- car, b- bus, t-two wheeler..etc) and number of parking hours then calculate the charges as given: bus- 20 rupees per hour, car- 10 rupees per hour, two wheelers- 5 rupees per hour.

**Solution:**

| #include<stdio.h> int main(){  char veh\_type;  int park\_hrs, charges;  scanf("%c", &veh\_type);  scanf("%d", &park\_hrs);  switch(veh\_type){  case 'b': charges = 20\*park\_hrs;  printf("Charges = %d Rs\n", charges);  break;  case 'c': charges = 10\*park\_hrs;  printf("Charges = %d Rs\n", charges);  break;  case 't': charges = 5\*park\_hrs;  printf("Charges = %d Rs\n", charges);  break;  }  return(0); } |
| --- |
|  |

3. C program to accept an array of elements as input (numeric or character).

1. Pass this array by pointer to a function.
2. Inside the function, exchange the first character with last character, second with last but one and so on.
3. Print the elements of the array after exchange in the main program.

**Solution:**

| #include<stdio.h> void reverse(int\* a, int sz){  for(int i = 0; i < sz -1 - i; i++){  int temp;  temp = a[i];  a[i] = a[sz - 1 - i];  a[sz - 1 - i] = temp;  } } int main(){  int a[10];  for(int i = 0; i < 10; i++)  scanf("%d", &a[i]);  reverse(a, 10);  for(int i = 0; i < 10; i++)  printf("%d ", a[i]);  return(0); } |
| --- |
|  |

4. Pooja would like to withdraw X $US from an ATM. The cash machine will only accept the transaction if X is a multiple of 5, and Pooja's account balance has enough cash to perform the withdrawal transaction (including bank charges). For each successful withdrawal the bank charges 0.50 $US.

Calculate Pooja's account balance after an attempted transaction.

### Input

Positive integer 0 < X <= 2000 - the amount of cash which Pooja wishes to withdraw.

Nonnegative number 0<= Y <= 2000 with two digits of precision - Pooja's initial account balance.

### Output

Output the account balance after the attempted transaction, given as a number with two digits of precision. If there is not enough money in the account to complete the transaction, output the current bank balance.

**Solution:**

| #include<stdio.h> int main(){  int withdraw\_amt;  float balance;  scanf("%d", &withdraw\_amt);  scanf("%f", &balance);  if( (withdraw\_amt % 5 == 0) && (balance + 0.5 >= withdraw\_amt))  balance -= withdraw\_amt + 0.5;  printf("%.2f\n", balance);  return(0); } |
| --- |

5. The Siruseri Sports Club organises an annual billiards game where the top two players of Siruseri play against each other. The Manager of Siruseri Sports Club decided to add his own twist to the game by changing the rules for determining the winner. In his version, at the end of each round, the cumulative **score** for each player is computed, and the leader and her current lead are found. Once all the rounds are over the player who had the maximum lead over all the rounds in the game is declared the winner.

Consider the following score sheet for a game with 5 rounds (input to the program):

| **Round** | **Player 1** | **Player 2** |
| --- | --- | --- |
| 1 | 140 | 82 |
| 2 | 89 | 134 |
| 3 | 90 | 110 |
| 4 | 112 | 106 |
| 5 | 88 | 90 |

The cumulative scores of both players, the leader and the lead after each round for this game is given below:

| **Round** | **Player 1** | **Player 2** | **Leader** | **Lead** |
| --- | --- | --- | --- | --- |
| 1 | 140 | 82 | Player 1 | 58 |
| 2 | 229 | 216 | Player 1 | 13 |
| 3 | 319 | 326 | Player 2 | 7 |
| 4 | 431 | 432 | Player 2 | 1 |
| 5 | 519 | 522 | Player 2 | 3 |

Note that the above table contains the cumulative scores.

The winner of this game is Player 1 as he had the maximum lead (58 at the end of round 1) during the game. This is the output of the program. Assume there are no ties.

**Solution:**

| #include<stdio.h> int main(){  int a[5], b[5], cuma, cumb, lead, lead\_player, maxlead, maxlead\_player;  maxlead = 0;  maxlead\_player = 1;  cuma = cumb = 0;  for(int i = 0; i < 5; i++){  printf("Enter scores of Player 1 and Player 2 at the end of Round %d ", i+1);  scanf("%d%d", &a[i], &b[i]);  cuma += a[i];  cumb += b[i];  if(cuma > cumb){  lead = cuma - cumb;  lead\_player = 1;  }  else if(cumb > cuma){  lead = cumb - cuma;  lead\_player = 2;  }  if(lead > maxlead){  maxlead = lead;  maxlead\_player = lead\_player;  }  }  printf("The winner is Player %d with lead %d", maxlead\_player, maxlead);  return(0); } |
| --- |

6. Mohan is very amazed by the find string functionality in his editor, and out of curiosity, he wants to write his own find string program to find if one string is substring of another. Help him to write such a program. Assume entered strings have no whitespace. You can make use of predefined string functions.

Input:

toogoodtobetrue

good

Output:

String good is a substring of toogoodtobetrue

**Solution:**

| #include<stdio.h> #include<string.h> int main(){  char s1[100], s2[100];  scanf("%s", s1);  scanf("%s", s2);  if(strstr(s1, s2))  printf("%s is a substring of %s\n", s2, s1);   else  printf("%s is not a substring of %s\n", s2, s1);   return(0); } |
| --- |

7. Y2K problem is that dates of the form ‘29/10/00’ become ambiguous because they could mean ‘29/10/1900’ or ‘29/10/2000’. Write a program that takes a sentence as argument (possibly containing spaces) with a single date at the beginning of the string, and replaces the date of the form ‘dd/mm/yy’ with ‘dd/mm/19yy’. Make use of predefined string functions.

**Solution:**

| #include<stdio.h> #include<string.h> int main(){  char buf[100], dst[100];  gets(buf);  strncpy(dst, buf, 6);  dst[6] = '1'; dst[7] = '9';  strcpy(dst+8, buf+6);  prints("%s\n", dst);  return(0); } |
| --- |

8. Jack is a teacher in a college. He frequently is in need of adding up the marks of all his students and calculating the average. Write a program that keeps on reading marks of students until -1 is entered, whereupon, the program prints the average (-1 is not included as a mark) and stops.

Input:

55 74 82 94 66 -1

Output:

74.2

**Solution:**

| #include<stdio.h> int main(){ int inp = 1;  float sum = 0; int count; int x[100]; for(count = 0: inp != -1; count++){  scanf("%d", &inp);  sum += inp; } if(count > 1) printf("%f\n", sum / (count -1 )); else printf("No input\n"); return(0); } |
| --- |
|  |

9. Ravi and Suresh are in a team writing a program. They find that the program calculates sum of arrays repeatedly a large number of times, and in order to have concise code, they want to write a function

int sum(int a[], int sz)

that will sum up an array of size ‘sz’. Demonstrate using a program how they can write and use such a function.

Input

5

55 74 82 94 66

Output:

371

**Solution:**

| #include<stdio.h> int sum(int a[], int sz){  int sum = 0; for(int i = 0; i < sz; i++)  sum += a[i]; return(sum); } int main(){  int n; scanf("%d", &n);  int myarr[n];  for(int i = 0; i < n; i++)  scanf("%d", &myarr[i]);  printf("%d\n", sum(myarr, n));  return(0); } |
| --- |

10. Hamid wants to write a program to read a line of string which can include space and tab characters (except newline) and print the number of whitespace characters in the line. Write a program to help him do so.

**Solution:**

| #include<stdio.h> int main(){  char line[256];  gets(line);  int countws = 0;  for(int i = 0; line[i] != '\0'; i++)  if((line[i] == ' ') || (line[i] == '\t'))  countws++;  printf(" Number of whitespace characters = %d\n", countws);  return(0); } |
| --- |

Alternatively getchar() can also be used.