Solution (Model Answer) to IAT-2 Feb 2022-Chem Cycle
Problem Solving through Programming(PSP)-21PSP13

1a. Define type conversion. Explain its types with suitable examples.

Ans.

Type conversion concept in C language is used to modify a variable from one date type to another data type. New data type should be mentioned before the variable name or value in brackets which to be typecast.

Example:

result = (float) 20/3;

- It is best practice to convert lower data type to higher data type to avoid data loss.
- Data will be truncated when higher data type is converted to lower. For example, if float is converted to int, data which is present after decimal point will be lost

Type conversion/Type Casting can be done 2ways:

a) Implicit Type conversion / Coercion / Automatic type conversion

It is done by compiler automatically:

example:

printf("%f", 20/3.00);

b) Explicit Type conversion / A cast

- **By preceding the expression with type in parenthesis**
- **It may be checked, unchecked or bit pattern.**

Syntax:

(type) expression;

Example:

result=(float) 20/3;

The result will be 6.666667

1b. Evaluate the below expression.

a+2>b||!c&&a==d||a-2<=e where a=11,b=6,c=0,d=7 and e=5.

Ans.

According to precedence of operation first arithmetic then relational then logical

Putting the values of variables a,b,c,d &e the expression will be:

```
11 + 2 > 6 || !0 && 11 == 7 || 11-2 <= 5
  13   >  6 || !0 && 11 == 7 || 9 <= 5
         1  ||  1  && 0 || 0
              1 || 0 || 0   = 1
```

So, answer is 1

2a. Describe the formatted input output functions with syntax and suitable example.

Ans:

printf() – Library function for formatted output:

Output data can be written on to a standard output device using the library function printf().

The printf statement provides certain features to control the alignment and spacing of printouts on the terminals. The general form of printf statement is:

printf ("control string", arg1,arg2, ..., argn);

Example: prinft("%d\t%s\t%d\n", roll, name, marks);

Control string of printf function consists of three types of item :

- Character that will be printed on the screen as they appear.
- Format specification that define the output format for display of each item.
- Escape sequence characters such as \n,\t, and \b can be in format specification.

scanf() – Library function for formatted input

Formatted input refers to an input data that has been arranged in a particular format. C library function scanf () is used to read/scan data from standard input device (keyboard). In general terms, scanf function is written as

scanf ("control string", &arg1, &arg2, ..., &argn);

ExampeL: scanf("%d%s%d", &roll, name, &marks);

The control string specifies the field format in which the data is to be entered and the arguments arg1,arg2.....,argn specify the address of locations where the data is stored. Control string contains field specification which direct the interpretation of input data. It may include:

- Field (or format) specification, consisting of the conversion character %, a data type character (or type specifier, and an optional number, specifying the field width.
- Blanks, tabs, or new lines.
- The arguments are written as variables preceded by & address of operator except arrays. array name itself represents address of first character.

2b. Differentiate between while and do while loop in C with example.

Ans.

- while is a pre-tested (entry-controlled) loop. Condition is checked before entering in the loop.
- do-while is a post-tested (exit-controlled) loop. Condition is checked after execution of the statements within the loop.

- The main difference is that do-while loop must execute once (because condition is checked at the end.)

while statement syntax:

```
while (condition)
{
        statement1;
        statement2;
        .....
}
```

/*working example – to display squares of numbers from 1 to 20 */

```
#include <stdio.h>
int main()
{
    int x;
    x=1;
    while(x<=20)
    {
            printf("%d  %d\n", x, x*x);
            x++;
```

```
        }
        return (0);
}
```

do…while statement syntax: (It is a post tested loop & must executes at least once)

```
    do
    {
        statement1;
        statement2;
        ......
    } while (condition);
```

```
/*working example – to display squares of numbers from 1 to 20 */
#include <stdio.h>
int main()
{
        int x;
        x=1;
        do
        {
                printf("%d  %d\n", x, x*x);
                x++;
        }while(x<=20);
        return (0);
}
```

3a. Define switch statement of C with syntax and example.

Ans.

switch statement Syntax:

```
switch (expression)
{
   case  condition1
            statement1;
            statement2;
            ......
            break;
    case condition2
            statement1;
            statement2;
            .......
            break;
    .....
    default:
            statement1;
            statement2;
            .......
    }
```

```
/* switch statement Example: Arithmetic operation by switch */
int main( )
{
```

```
        float n1,n2,result;
        int choice;
        printf("Enter 2 numbers : ");
        scanf("%f%f", &n1, &n2);
          printf("1. Addition 2. Subtraction 3. Multiplication 4. Division:\n");
        printf("Enter your choice number : ");
        scanf("%d", &choice);
        switch(choice)
        {
                case 1 : result = n1+n2;
                         break;
                case 2 : result = n1-n2;
                         break;
                case 3 : result = n1*n2;
                         break;
                case 4 : if(n2 == 0)
                                 { printf("divide error!!!\n"); return (99); }
                         else
                                 result = n1/n2;
                        break;
                default: printf("Wrong choice!\n");
        }
        printf("Result = %f\n", result);
        return (0);
}
```

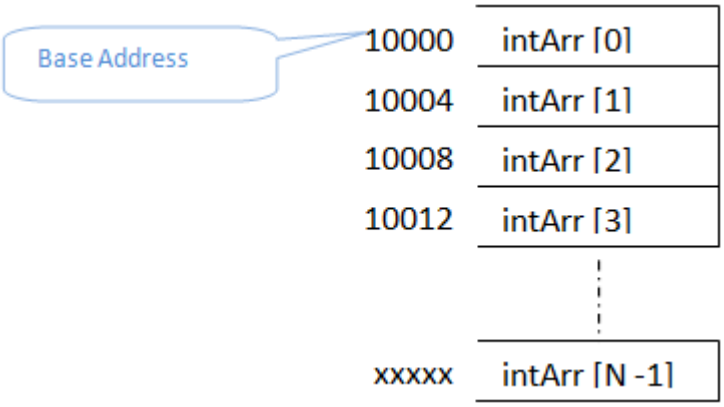
Expected Output:
Enter 2 numbers : 50 25
1. Addition 2. Subtraction 3. Multiplication 4. Division:
Enter your choice number : 4
Result = 2.000000

3b. Define an Array. How 1 dimension integer array is represented in memory. With the help of suitable examples demonstrate the array initialization.

Ans.

An array is an identifier that refers to the collection of data items of same type which all have the same name. The individual data items are represented by their corresponding array elements.

Memory representation of the array:
Below diagram shows how memory is allocated to an integer array of N elements. Its base address – address of its first element is 10000. Since it is an integer array, each of its element will occupy 4 bytes of space (earlier 2 bytes for integer).



Address of an element (subscript) ranges from 0 to n-1, where n is total number of elements.
Declaration of one-dimensional array:
One dimensional array:
syntax:

datatype arrayname[size];

Examples:

```
float height[50];
int batch[10];
char name[20];
```

Initialization of one-dimensional array:
Examples:

```
int marks[5] = {76,45,-12,87,92};
int a[5] = {0};
/* if one element is initialized, rest of the elements will be initialized to zero */
int a[5] = {1};
/* first element a[0] is  = 1, rest all will be initialized with 0 */
```

Initializing by Program:

```
#include <stdio.h>
int main() {
   int x, arr[10]={2,0,3};
  printf("Array Elements are :\n");
  for(x=0;x<10;x++)
      printf("%4d",arr[x]);
  printf("\n");
  return (0);
}
```

4. Define two-way selection statement. Explain if, if_else, nested_if and ladder else_if with suitable syntax flowchart and example.
Ans.

Conditional/Selection Branching/constructs/2-way selection
In C programming language branching statements are: if, if else & switch
if/if else can be extended to: ladder if & nested if

if statement:

Syntax:
if (expression/condition)

```
{
   statement1;
   statement2;
   ...
 }
```

if … else statement:

Syntax:

```
if (expression/condition) {
   statement1;
   statement2;
   ...
 }
else {
   statement1;
   statement2;
   ...
 }
#include <stdio.h>
int main()
{
    int age;
    printf("Enter your age : ");
    scanf("%d", &age);
    if(age >= 25)
        printf("Celebrate valentine day\n");
    else
        printf("Just keep looking cute only.\n");
    return (0);
}
```

Nested if

```
if (expression/condition)
{
        if (expression/condition)
         {
                statement1;
                statement2;
                .....
        }
        else
        {
                statement1;
                statement2;
                .....
        }
 }
 else
 {
         if (expression/condition)
```

```
        {
                statement1;
                statement2;
                 .....
        }
         else
        {
                statement1;
                statement2;
                .....
       }
}

// Example: to find maximum in 3 numbers
#include <stdio.h>
int main()
{
    int n1,n2,n3,max;
    printf("Enter 3 numbers : ");
    scanf("%d%d%d", &n1, &n2, &n3);
    if(n1>n2)
        if(n1>n3)
            max=n1;
        else
            max=n3;
    else
        if(n2>n3)
            max=n2;
        else
            max=n3;

    print("Maximum = %d\n", max);
    return (0);
}
```

if … else ladder

```
if (expression/condition)
{
  //statements;
 }
  else  if (expression/condition)
   {
        //statements;
    }
   else  if (expression/condition)
     {
        //statements;
     }
        ……
       else
```

```
        {
         //statements;
         }
```

```
// Example: to compute grade based on marks
#include <stdio.h>
int main()
{
int marks; char grade;
printf("Enter marks : ");
scanf("%d", &marks);
if(marks >= 90)
    grade='S';
    else if (marks>=80)
        grade='A';
     else if (marks>=70)
         grade='B';
      else if (marks >=60)
          grade='C';
        else if (marks>=50)
          grade='D';
          else if (marks>=40)
              grade='D';
           else
              grade='F';
printf("Grade is %c\n", grade);
return (0);
}
```

5. Write c program to find the roots of the quadratic equation with suitable messages.

Ans.

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

int main()
{
   double a, b, c, d, r1, r2, real, imag;
   printf("\nEnter coefficients a, b and c: ");
   scanf("%lf %lf %lf",&a, &b, &c);
   d = b*b-4*a*c;  /* discriminant */

 if (d == 0)    {
     r1 = r2 = -b/(2*a);
     printf("Roots are real : root1 = root2 = %.2lf;", r1);
  }
 else
   if (d > 0)      {
```

```
        r1 = (-b+sqrt(d))/(2*a);
           r2 = (-b-sqrt(d))/(2*a);
        printf("Roots are distinct: root1 = .2lf root2 = %.2lf\n",r1, r2);
       }
     else    {
        real = -b/(2*a);
           imag = sqrt(-d)/(2*a);
        printf("Roots are imaginary:\n");
        printf("root1 = %.2lf+%.2lfi and root2 = %.2lf-%.2lfi", real, imag, real, imag);
     }
     return (0);
}
```

Output:
Enter coefficients a, b and c: 2 4 2
Roots are real : root1 = root2 = -1.00;
Enter coefficients a, b and c: 2 2 2
Roots are imaginary:
root1 = -0.50+0.87i and root2 = -0.50-0.87i (May be blank if no time to calculate)

6a. Write a C program to check whether a number is palindrome or not.
Ans.

```
/* Reverse of a number & whether the number is palindrome */
#include<stdio.h>
void main()
{
   int num, rev=0, temp;
   printf("\nEnter the number: ");
   scanf("%d", &num);
   temp = num;
   while(temp != 0){
    rev=rev*10 + temp%10;
    temp/=10;
   }
   printf("The Reverse of the number is %d\n", rev);
   if (rev==num) printf("The number is a palindrome.\n");
   else printf("The number is not a palindrome.\n");
}
```

Expected Outputs:
Enter the number: 12321
The Reverse of the number is 12321
The number is a palindrome.
Enter the number: 4567
The Reverse of the number is 7654
The number is not a palindrome.

6b. Demonstrate the working of break and continue statements with examples.

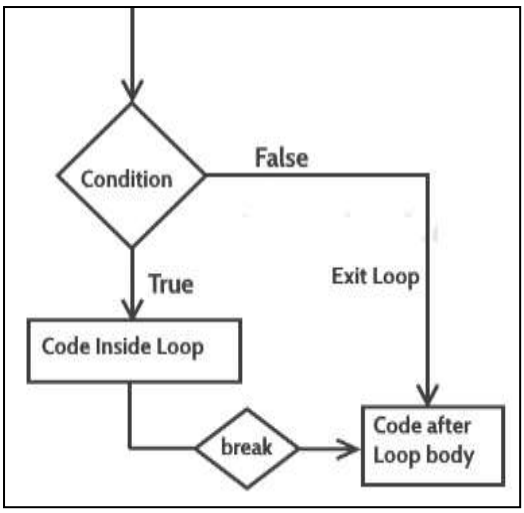
Ans.

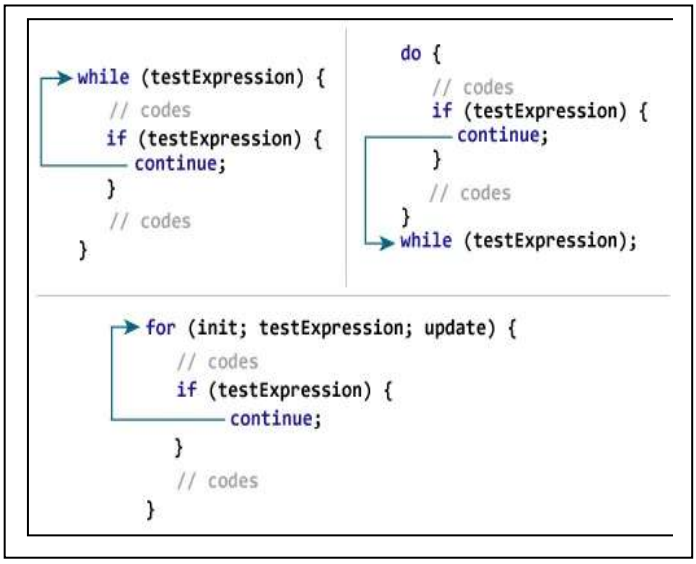
- The break is a keyword in C which is used to bring the program control out of the loop. The break statement is used inside loops or switch statement.
- The break statement breaks the loop one by one, i.e., in the case of nested loops, it breaks the inner loop first and then proceeds to outer loops.
- break statement would only exit from the loop containing it.

```
/*Example program to check a number whether it is prime with break */
#include<stdio.h>
#include <math.h>
int main()
{
        int n,d,prime=1;
        printf("Enter a number : ");
        scanf("%d", &n);
        for (d=2;d<sqrt(n);d++)
        {
                if (n%d==0) {   prime=0;        break; /* there is no need to check further */
        }
        if (prime)    printf("Yes %d is a prime number.\n", n);
        else    printf("No, %d is not a prime number.\n", n);
        return (0);
}
```

- The continue statement is used in loops to skip the following statements in the loop and to continue with the next iteration of the loop

```
#include<stdio.h>
int main()
{
  int marks,sum=0, x;
  for (x=1;x<=6;x++) {
      printf("Enter marks %d : ", x);
      scanf("%d", &marks);
      if(marks <0 || marks > 100)
      {
              printf("Invalid marks!!!\n");
              x--;
            continue; /* again read marks for same paper */
      }
      sum+=marks;
   }
printf("sum of marks = %d.\n", sum);
return (0);
}
```





7a. Write a C program to plot Pascal's triangle.

Ans.

```
/* Pascal's triangle - Dr. P. N. Singh */
#include <stdio.h>
int main() {
  int x,y,z,p;
  for(x=0;x<9;x++)    {
      for(y=1;y<20-x;y++)
           printf("  "); /* for blank spaces */
       for(y=0;y<x;y++) {
            if(x==0 || y== 0)  p=1;
            else  p=p*(x-y)/y;       /* For Binomial co-efficient */
            printf("%4d",p);      /* printing value taking 4 minimum spaces */
       }
       printf("\n");
  }
  return (0);
}
```

Expected output: Plotting Pascal's triangle

```
            1
          1   1
        1   2   1
      1   3   3   1
    1   4   6   4   1
  1   5  10  10   5   1
1   6  15  20  15   6   1
1   7  21  35  35  21   7   1
```

7b. Explain 5 Bit-wise operators with example

Ans.

& - Bit-wise AND : 1 only when both inputs are 1

printf(“%d”, 20 & 25); /*10100 & 11001 = 10000 will give output 16*/
| - Bit-wise or : 1 when any of the input is 1. It gives 0 when both inputs are 0
printf(“%d”, 20 | 25); /*10100 | 11001 = 11101 will give output 29*/
^ - Bit wise XOR (Exclusive or): 1 when only one input is 1. If both are 1 or 0, it gives 0
printf(“%d”, 20 ^ 25); /*10100 ^ 11001 = 01101 will give output 13*/
~ - Bit-wise compliment : Difference in highest & given. In Binary 1 will be 0 and 0 will be 1. If unsigned int has 2 bytes (16 bits) size with highest value 65535 then
printf(“%u”, ~ 5); /*~ 0000000000000101 = 1111111111111010 will give output 65530*/
>> - Bit wise right shift : Halves the value in integer
printf(“%u”, 10>>2); /*1010 will 101 in first right shift & again 10 in 2nd right shift so 2 */
<< - Bit wise left shift : doubles the value(within range)
printf(“%u”, 11<<2); /*1011 will 10110 in first left shift & again 101100 in 2nd left shift so 44 */