| Question No. 01 |
| --- |
| Declare a structure called Time that will hold three integers hour, minute, and second. Declare a Time variable that holds 6:30 PM. (Think about 24 hour Time Format) |
| #include <stdio.h>  struct Time  {  int hour;  int minute;  int second;  };  int main()  {  struct Time time = {18, 30, 00};  //printf("%d %d %d", time.hour, time.minute, time.second);  return 0;  } |

| Question No. 02 |
| --- |
| Consider the following structure that represents a time interval:  struct Interval{  struct Time start;  struct Time end;  }  Declare an interval that starts at 5:30 and ends at 10:15. |
| #include <stdio.h>  struct Time  {  int hour;  int minute;  int second;  };  struct Interval  {  struct Time start;  struct Time end;  };  int main()  {  struct Interval i;  i.start.hour = 5;  i.start.minute = 30;  i.start.second = 00;  i.end.hour = 5;  i.end.minute = 30;  i.end.second = 00;  } |

| Question No. 03 |
| --- |
| Can a struct have a member variable that is an array? Give an example. |
| Yes.  struct Student  {  char name[100];  int class;  int roll;  struct Date dob;  }; |

| Question No. 04 |
| --- |
| Remember the fraction structure that we wrote. Write a function that takes a fraction and returns its inverse. For example, the inverse of 2/3 is 3/2. |
| #include <stdio.h>  struct Fraction  {  int num;  int denom;  };  struct Fraction reverse(struct Fraction a)  {  struct Fraction f;  f.num = a.denom;  f.denom = a.num;  return f;  }  struct Fraction inputF()  {  struct Fraction f;  scanf("%d %d", &f.num, &f.denom);  return f;  }  void print(struct Fraction f)  {  printf("%d/%d", f.num, f.denom);  }  int main()  {  struct Fraction a, b;  a = inputF();  b = reverse(a);  print(a);  printf("\n");  print(b);  return 0;  } |

| Question No. 05 |
| --- |
| Remember the student structure we wrote.  struct Student {  char name[100];  int roll;  int class;  struct Date dob;  }  Show how you can update the name and roll of a student. |
| struct Student update(struct Student f)  {  char s[100];  scanf(" %s",s);  strcpy(f.name, s);    int roll;  scanf("%d",&roll);  f.roll = roll;  return f;  } |

| Question No. 06 |
| --- |
| How can you use a structure to return multiple values from a function. Give an example. |
| #include <stdio.h>  struct Result{  int vagfol;  int vagshesh;  };  struct Result div(int x, int y){  return (struct Result){x / y, x % y};  }  int main()  {  int a = 7, b = 2;  struct Result result = div(a, b);  printf("\n%d %d", result.vagfol, result.vagshesh);  } |

| Question No. 07 |
| --- |
| Find the binary representation of the number 23. |
| #include <stdio.h>  #include <stdbool.h>  #include <string.h>  void ToBinary(unsigned int val, char str[])  {  int idx = 0;  while (val)  {  int rem = val % 2;  val = val / 2;  str[idx] = '0' + rem;  idx++;  }  str[idx] = '\0';  strrev(str);  }  int main()  {  char str[1001];  unsigned int v = 23;  ToBinary(v, str);  puts(str);  return 0;  } |

| Question No. 08 |
| --- |
| Write a program that takes input two integers l and r and finds the xor of all numbers between l and r. |
| #include <stdio.h>  #include <stdbool.h>  #include <string.h>  void ToBinary(unsigned int val, char str[], int k)  {  int idx = 0;  while (val)  {  int rem = val % 2;  val = val / 2;  str[idx] = '0' + rem;  idx++;  }  while ((idx + 1) != k)  {  str[idx] = '0';  idx++;  }  str[idx] = '\0';  strrev(str);  }  int main()  {  char sx[1001], sy[1001], sxor[1001];  unsigned int x = 25;  unsigned int y = 10;  ToBinary(x, sx, 8);  puts(sx);  ToBinary(y, sy, 8);  puts(sy);  unsigned int xor = x ^ y;  ToBinary(xor, sxor, 8);  printf("%s\t%d\n", sxor,xor);  return 0;  } |

| Question No. 09 |
| --- |
| Suppose you want to flip the leftmost one bit of a number. For example, 00101100 would become 00001100. The leftmost 1 bit became a zero. Write a program to do that. |
| #include <stdio.h>  #include <stdbool.h>  #include <string.h>  struct Binary  {  char str[50];  };  unsigned int ToInt(char str[])  {  int len = strlen(str);  unsigned int ans = 0;  for (int i = 0; i < len; i++)  ans = ans \* 2 + str[i] - '0';  return ans;  }  struct Binary ToBinary(unsigned int val, int k)  {  struct Binary ans;  int idx = 0;  while (val)  {  int rem = val % 2;  val = val / 2;  ans.str[idx] = '0' + rem;  idx++;  }  while ((idx) != k)  {  ans.str[idx] = '0';  idx++;  }  ans.str[idx] = '\0';  strrev(ans.str);  return ans;  }  bool getBit(unsigned int mask, int k)  {  return (mask & (1 << k)) != 0;  }  unsigned int clearBit(unsigned int mask, int k)  {  return mask & ~(1 << k);  }  int main()  {  char str[1001];  gets(str);  unsigned int bin = ToInt(str);  // printf("%u\n", bin);  unsigned int x = bin;  ToBinary(x, 8);  // puts(str);  // printf("%d = %s\n", x, ToBinary(x, 8).str);  for (int i = 7; i >= 0; i--)  {  if (getBit(x, i) == 1)  {  printf("Flipping Leftmost 1 bit: %s\n", ToBinary(clearBit(x, i), 8).str);  break;  }  }  return 0;  } |

| Question No. 10 |
| --- |
| Suppose you want to make the last 4 bits of an integer 0. For example 11010011 becomes 11010000. How can you do it? Can you do it with only 2 bitwise operations? |
| #include <stdio.h>  #include <stdbool.h>  #include <string.h>  struct Binary{  char str[50];  };  unsigned int ToInt(char str[]){  int len = strlen(str);  unsigned int ans = 0;  for (int i = 0; i < len; i++)  ans = ans \* 2 + str[i] - '0';  return ans;  }  struct Binary ToBinary(unsigned int val, int k){  struct Binary ans;  int idx = 0;  while (val)  {  int rem = val % 2;  val = val / 2;  ans.str[idx] = '0' + rem;  idx++;  }  while ((idx) != k)  {  ans.str[idx] = '0';  idx++;  }  ans.str[idx] = '\0';  strrev(ans.str);  return ans;  }  unsigned int clearBit(unsigned int mask, int k){  return mask & ~(1 << k);  }  int main()  {  int count =0;  char str[1001];  gets(str);  unsigned int x = ToInt(str);  ToBinary(x, 8);  for (int i = 0; i < 8; i++)  {  if (count<4)  {  x = clearBit(x, i);  count++;  }  }  printf("%s\n", ToBinary(x, 8).str);  return 0;  } |

**Question Paper:** [Theory Exam 6](https://docs.google.com/document/d/1pI8dYh143MYf4oJZNmSt2Tl4m5QNTwIVzvpQETkdi44/edit)