Mata Kuliah : Konsep Pemrograman ( Kelas B ) Semester : Ganjil 2021/2022

Pengampu : Drs. Bambang Harjito, M.App.Sc, PhD Waktu : --

Ujian : Evaluasi ke 1 Hari, Tanggal : Sept 2022

1. **Hasil pekerjaan berupa file .doc atau .pdf dan jadikan satu file selanjutnya Upload ke Classroom**
2. **Beri nama file : NIM\_nama\_depan**
3. **Tidak boleh mengupload berupa gambar atau ZIP**

Soal 1. Jawaban singkat [Bobot : 25%]

Jelaskan maksud instruksi-instruksi dalam bahasa C++ di bawah ini. Instruksi ini bisa muncul di bagian mana pun dari suatu program



Berikut adalah intruksi yang harus dijelaskan

|  |  |
| --- | --- |
| a | float arr[5]; |
| b | void instructions(void); |
| c | double fractions[SIZE] |
| d | **int** sum(**int** x, **int** y)  {  **int** return = x + y; } |
| e | **for** (x = **0**; x < **SIZE**; ++x) {  **for** (y = **0**; y < **SIZE**; ++y) {  printf(**"table[%d][%d] = %d\n"**, x, y, table[x][y]);  }  } |
| **Tuliskan penjelasan instruksi-instruksi di atas di bagian kosong di bawah ini** | |
| 1.a  1.b  1.c  1.d  1.e | Instruksi untuk mendeklarasikan array ber-type float dengan nama arr yang mempunyai panjang 5.  Instruksi untuk mendeklarasikan fungsi yang bernama instructions dengan type fungsi void, sehingga tidak memiliki nilai kembalian (*return value*).  Instruksi untuk mendeklarasikan array ber-type double dengan nama fractions yang panjangnya sama dengan variable SIZE yang telah di-define sebelumnya.  Sebuah fungsi type integer yang bernama sum dengan variabel x dan y sebagai parameternya, fungsi tersebut me-return (mengembalikan nilai) variabel x + y;  Melakukan perulangan (for) dengan menambahkan variabel i sampai dengan variabel SIZE yang telah di-define sebelumnya, didalam for tersebut melakukan perulangan lagi dengan menambahkan variable y sampai dengan SIZE. Didalamnya dicetak array bernama table dengan index ke-[x][y]. |

**2. Soal 2. Hasil Eksekusi Program [Bobot : 25%]**

**Tuliskan keluaran (apa yang tercetak di layar) dari program-program di bawah ini pada kotak koson**g di samping kanan program. Jika tidak ada keluaran, tuliskan: *Tidak ada keluaran*. Semua komentar program sengaja dihilangkan

|  |  |  |
| --- | --- | --- |
| **No** | **Program** | **Keluaran** |
| 1 | **#include<stdio.h>**  **int main(){**  **double num=5.2;**  **int  var=5;**  **printf("%d\t",sizeof(!num));**  **printf("%d\t",sizeof(var=15/2));**  **printf("%d",var);     return 0;**  **}** | 4 4 5 |
| 2 | **#include<stdio.h>**  **int main(){**  **int i=5,j;**  **j=++i+++i+++i;**  **printf("%d %d",i,j);**  **return 0;**  **}** | Tidak ada keluaran (compilation error) |
| 3 | **#include<stdio.h>**  **int main(){**  **int a=2;**  **a=a++ + ~++a;**  **printf("%d",a);**  **return 0;**  **}** | -3 |
| 4 | **#include<stdio.h>**  **#define p(a,b) a##b**  **#define call(x) #x**  **int main(){**  **do{**  **int i=15,j=3;**  **printf("%d",p(i-+,+j));**  **}**  **while(\*(call(625)+3));**  **return 0;**  **}** | 11 |
| 5 | **#define PRINT**  **printf("Star Wars");**  **printf(" Psycho");**  **#include<stdio.h>**  **void main(){**  **int x=1;**  **if(x--)**  **PRINT**  **else**  **printf("TheShawshank Redemption");**  **}** | Tidak ada keluaran (compilation error) |
| 6 | **#include<stdio.h>**  **int main(){**  **int arr[]={1,2,5,7,8,0,11,5,50};**  **int i,avg;**  **for(int i=0;i<12;i++){**  **avg=avg+arr[i];**  **}**  **printf("%d",avg/12);     return 0;**  **}** | 1311651 |

**Soal 3. Fungsi , Prosedur dan Array [Bobot : 45%], kerjakan**

|  |  |
| --- | --- |
| 3.1 | (Coin Tossing) Write a program that simulates coin tossing. For each toss of the coin the program should print Heads or Tails. Let the program toss the coin 100 times, and count the number of times each side of the coin appears. Print the results. The program should call a separate function flip that takes no arguments and returns for tails and 1 for heads. [Note: If the program realistically simulates the coin tossing, then each side of the coin should appear approximately half the time for a total of approximately 50 heads and 50 tails.]  **Code :**  #include <stdio.h>  #include <stdlib.h>  int flip();  int main()  {      int tails = 0, heads = 0;      for (int toss = 0; toss < 100; toss++)      {          int x = flip();          printf("%d ", x);          if (x == 0)          {              tails++;          } else          {              heads++;          }      }      printf("\nMenghasilkan heads sebanyak: %d kali", heads);      printf("\nMenghasilkan tails sebanyak: %d kali", tails);  }  int flip()  {      int i = rand() % 2;      if(i == 0){          return 0;      } else {          return 1;      }  }  **Output :**  1 1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 1  Menghasilkan heads sebanyak: 48 kali  Menghasilkan tails sebanyak: 52 kali |
| 3.2 | (Palindromes) A palindrome is a string that’s spelled the same way forward and backward. Some examples of palindromes are: “radar,” “able was i ere i saw elba,” and, if you ignore blanks, “a man a plan a canal panama.” Write a recursive function testPalindrome that returns 1 if the string stored in the array is a palindrome and 0 otherwise. The function should ignore spaces and punctuation in the string  **Code :**  #include <stdio.h>  #include <string.h>  char input[100], newInput[100];  int palindromeCheck(), removeSpace();  int main(){    printf("Masukkan kalimat: ");    scanf(" %[^\n]", &input);    removeSpace(input);      if(palindromeCheck(newInput, 0, strlen(newInput)-1) == 1)    {        printf("Kalimat tersebut merupakan palindrome\n");    } else    {        printf("Kalimat tersebut bukan merupakan palindrome\n");    }  }  int removeSpace(char \*input)  {    int i = 0, j = 0;      while(input[i] != '\0')      {          if(input[i] == ' ')          {              input[i] = '\0';          }          else if (input[i] != ' ')          {              newInput[j] = input[i];              j++;          }          i++;      }  }  int palindromeCheck(char \*newInput, int start, int end)  {    int i = end/2;    if (end == i) {      return 1;    } else if (newInput[start] != newInput[end]) {      return 0;    } else {      return palindromeCheck(newInput, start+1, end-1);    }  }  **Output :**  Masukkan kalimat: a man a plan a canal panama  Kalimat tersebut merupakan palindrome |
| 3.3 | (The Sieve of Eratosthenes) A prime integer is any integer greater than 1 that can be divided evenly only by itself and 1. The Sieve of Eratosthenes is a method of finding prime numbers. It works as follows:a) Create an array with all elements initialized to 1 (true). Array elements with prime indices will remain 1. All other array elements will eventually be set to zero. b) Starting with array index 2 (index 1 is not prime), every time an array element is found whose value is 1, loop through the remainder of the array and set to zero every element whose index is a multiple of the index for the element with value 1. For array index 2, all elements beyond 2 in the array that are multiples of 2 will be set to zero (indices 4, 6, 8, 10, and so on.). For array index 3, all elements beyond 3 in the array that are multiples of 3 will be set to zero (indices 6, 9, 12, 15, and so on.). When this process is complete, the array elements that are still set to 1 indicate that the index is a prime number. Write a program that uses an array of 1,000 elements to determine and print the prime numbers between 1 and 999. Ignore element 0 of the array.  **Code :**  #include <stdio.h>  int main()  {      int angka[1000], i, a, jumlahPrima = 0;      for (i = 0; i < 1000; i++) {          angka[i] = 1;      }        for (i = 1; i < 1000; i++) {          if (angka[i] == 1 && i != 1) {              for (a = i; a <= 1000; a++) {                  if (a % i == 0 && a != i) {                      angka[a] = 0;                  }              }          }      }      for (i = 2; i < 1000; i++) {          if (angka[i] == 1) {              ++jumlahPrima;          }      }      printf("Angka prima dari 1 - 1000 yaitu sejumlah %d bilangan.\n", jumlahPrima);      return 0;  }  **Output :**  Angka prima dari 1 - 1000 yaitu sejumlah 168 bilangan. |

**Pernyataan Verifikasi**

Semua materi uji kompetensi dalam mata kuliah ini telah diverifikasi sesuai dengan materi kompetensi dasar pada mata kuliah yang bersesuai dalam Kurikulum Informatika th 2020

Pembuat Soal Verifikator Kaprodi

Drs Bambang Harjito, M App. Sc, Ph D DR Wiharto MKom